Gibbs_Sampling

November 2, 2018

1 1. Alogorithm Description

Our Alogorithm is as below:

1. Generate $x_0 \sim Uniform[0, B]$ $y_0 \sim Uniform[0, B]$

2. For t = 0 to (T - 1):
   2.1 Genertae $u_x \sim Uniform[0, 1]$ and set $x_{t+1} = -\frac{1}{y_t} log(1 - (1 - \exp(-y_t B))u_x)$
   2.1 Generate $u_y \sim Uniform[0, 1]$ and set $y_{t+1} = -\frac{1}{x_{t+1}} log(1 - (1 - exp(-x_{t+1} B))u_y)$

The implementation is shown below:

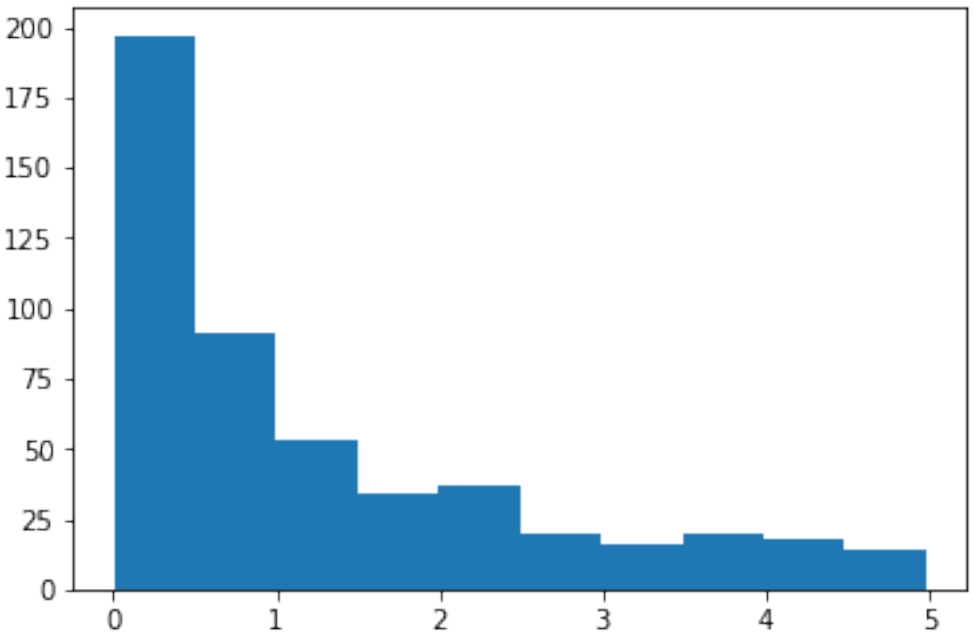
```
In [11]: import numpy as np
         import random as rand

         B = 5
         T = 500

         sample_record = np.zeros((T, 2))
         sample_record[0, 0] = rand.uniform(0, B)
         sample_record[0, 1] = rand.uniform(0, B)

         for t in np.arange(T-1):
             x_t = sample_record[t, 0]
             y_t = sample_record[t, 1]
             u_x = rand.uniform(0, 1)
             u_y = rand.uniform(0, 1)
             x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
             sample_record[t + 1, 0] = x_tadd1
             sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

         sample_record = np.matrix(sample_record)
```

2 2. Plot the histogram of values for x

```
In [12]: # T=  500
```

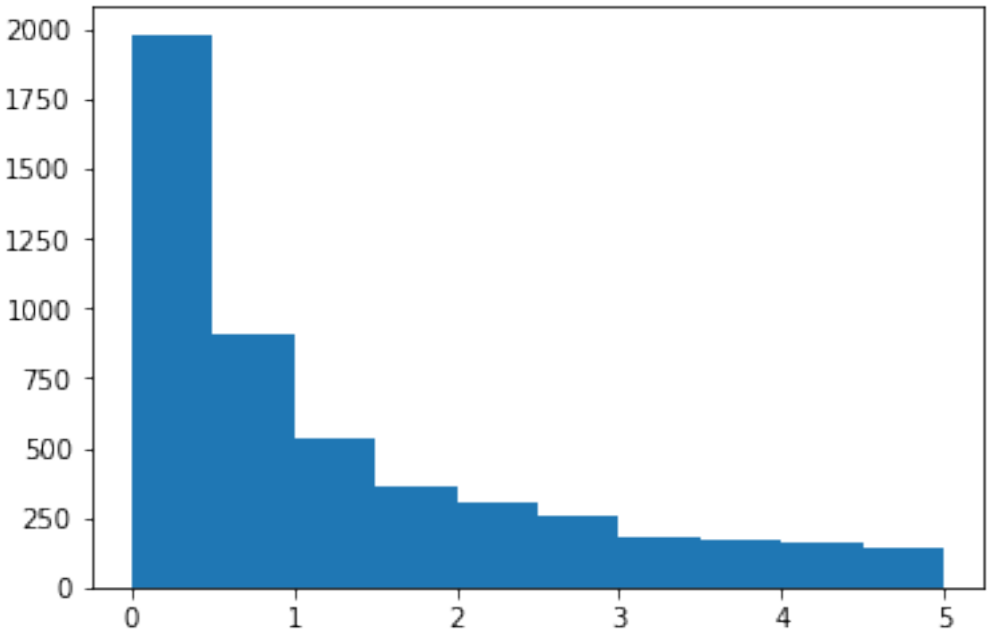
```
import numpy as np
import random as rand
import matplotlib.pyplot as plt

B = 5
T = 500

sample_record = np.zeros((T, 2))
sample_record[0, 0] = rand.uniform(0, B)
sample_record[0, 1] = rand.uniform(0, B)

for t in np.arange(T-1):
    x_t = sample_record[t, 0]
    y_t = sample_record[t, 1]
    u_x = rand.uniform(0, 1)
    u_y = rand.uniform(0, 1)
    x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
    sample_record[t + 1, 0] = x_tadd1
    sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

sample_record = np.matrix(sample_record)

%matplotlib inline
plt.hist(sample_record[:, 0])
```

```
Out[12]: (array([197.,  91.,  53.,  34.,  37.,  20.,  16.,  20.,  18.,  14.]),
          array([1.26835151e-03, 4.99508471e-01, 9.97748590e-01, 1.49598871e+00,
                 1.99422883e+00, 2.49246895e+00, 2.99070907e+00, 3.48894919e+00,
                 3.98718931e+00, 4.48542943e+00, 4.98366955e+00]),
          <a list of 10 Patch objects>)
```



In [13]:
```
# T=  5000

import numpy as np
import random as rand
import matplotlib.pyplot as plt

B = 5
T = 5000

sample_record = np.zeros((T, 2))
sample_record[0, 0] = rand.uniform(0, B)
sample_record[0, 1] = rand.uniform(0, B)

for t in np.arange(T-1):
    x_t = sample_record[t, 0]
    y_t = sample_record[t, 1]
    u_x = rand.uniform(0, 1)
    u_y = rand.uniform(0, 1)
    x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
    sample_record[t + 1, 0] = x_tadd1
    sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

sample_record = np.matrix(sample_record)

%matplotlib inline
plt.hist(sample_record[:, 0])
```

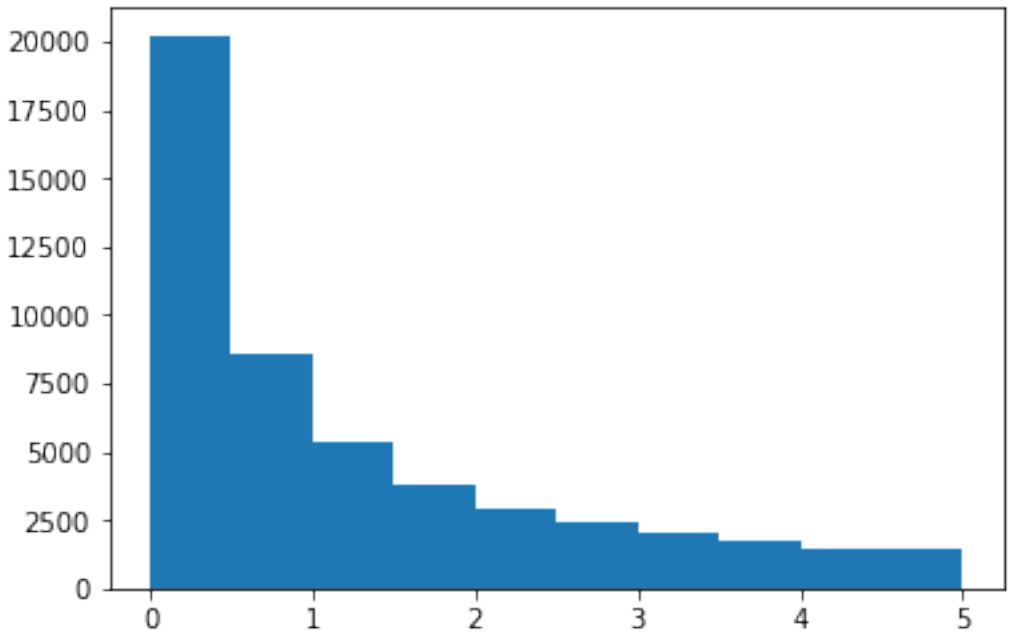
Out[13]: (array([1983., 903., 537., 366., 302., 255., 184., 175., 158.,
137.]),
array([1.34585478e-04, 5.00025549e-01, 9.99916512e-01, 1.49980747e+00,
1.99969844e+00, 2.49958940e+00, 2.99948036e+00, 3.49937133e+00,
3.99926229e+00, 4.49915325e+00, 4.99904422e+00]),
<a list of 10 Patch objects>)



In [14]:
```
# T=  50000

import numpy as np
import random as rand
import matplotlib.pyplot as plt

B = 5
T = 50000

sample_record = np.zeros((T, 2))
sample_record[0, 0] = rand.uniform(0, B)
sample_record[0, 1] = rand.uniform(0, B)

for t in np.arange(T-1):
    x_t = sample_record[t, 0]
    y_t = sample_record[t, 1]
    u_x = rand.uniform(0, 1)
    u_y = rand.uniform(0, 1)
    x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
```

```
        sample_record[t + 1, 0] = x_tadd1
        sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

    sample_record = np.matrix(sample_record)

    %matplotlib inline
    plt.hist(sample_record[:, 0])
```

Out[14]:
```
(array([20218.,  8536.,  5390.,  3766.,  2957.,  2446.,  2045.,  1745.,
         1473.,  1424.]),
 array([1.48639325e-05, 4.99974833e-01, 9.99934802e-01, 1.49989477e+00,
        1.99985474e+00, 2.49981471e+00, 2.99977468e+00, 3.49973465e+00,
        3.99969462e+00, 4.49965458e+00, 4.99961455e+00]),
 <a list of 10 Patch objects>)
```



3 3. Estimation of Expectation

In [15]:
```
# T = 500

import numpy as np
import random as rand

B = 5
T = 500
```

```
sample_record = np.zeros((T, 2))
sample_record[0, 0] = rand.uniform(0, B)
sample_record[0, 1] = rand.uniform(0, B)

for t in np.arange(T-1):
    x_t = sample_record[t, 0]
    y_t = sample_record[t, 1]
    u_x = rand.uniform(0, 1)
    u_y = rand.uniform(0, 1)
    x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
    sample_record[t + 1, 0] = x_tadd1
    sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

sample_record = np.matrix(sample_record)

estimated_expec = sample_record.mean(0)[0, 0]
estimated_expec
```

Out[15]: 1.1296581833637178

In [16]:
```
# T = 5000

import numpy as np
import random as rand

B = 5
T = 5000

sample_record = np.zeros((T, 2))
sample_record[0, 0] = rand.uniform(0, B)
sample_record[0, 1] = rand.uniform(0, B)

for t in np.arange(T-1):
    x_t = sample_record[t, 0]
    y_t = sample_record[t, 1]
    u_x = rand.uniform(0, 1)
    u_y = rand.uniform(0, 1)
    x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
    sample_record[t + 1, 0] = x_tadd1
    sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

sample_record = np.matrix(sample_record)

estimated_expec = sample_record.mean(0)[0, 0]
estimated_expec
```

Out[16]: 1.2242680849525611

In [17]:
```
## T = 50000
```

```
import numpy as np
import random as rand

B = 5
T = 50000

sample_record = np.zeros((T, 2))
sample_record[0, 0] = rand.uniform(0, B)
sample_record[0, 1] = rand.uniform(0, B)

for t in np.arange(T-1):
    x_t = sample_record[t, 0]
    y_t = sample_record[t, 1]
    u_x = rand.uniform(0, 1)
    u_y = rand.uniform(0, 1)
    x_tadd1 = -(1 / y_t) * np.log(1 - (1 - np.exp(-y_t * B)) * u_x)
    sample_record[t + 1, 0] = x_tadd1
    sample_record[t + 1, 1] = -(1 / x_tadd1) * np.log(1 - (1 - np.exp(- x_tadd1 * B))

sample_record = np.matrix(sample_record)

estimated_expec = sample_record.mean(0)[0, 0]
estimated_expec
```

Out[17]: 1.2619017909294812