hw5

April 2, 2021

[46]:
```
%config IPCompleter.use_jedi = False
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import cvxpy as cp
import pandas as pd
np.set_printoptions(precision=4)
from tqdm.notebook import tqdm

from pathlib import Path
fig_path = str(Path().absolute())+'/figures/hw5/'
print(fig_path)
# data_path = str(Path().absolute())+'/hw5/data/'
# quiz_data_path = str(Path().absolute())+'/hw5/quiz5_data/data/'
# print(data_path)
```

```
/home/zpyang/grad_courses/2021_spring/ece595_ml/figures/hw5/
```

1 Exercise 2

a) $\mu_1 = \mu_{min} = \mu_{rand} = \frac{1}{2}$

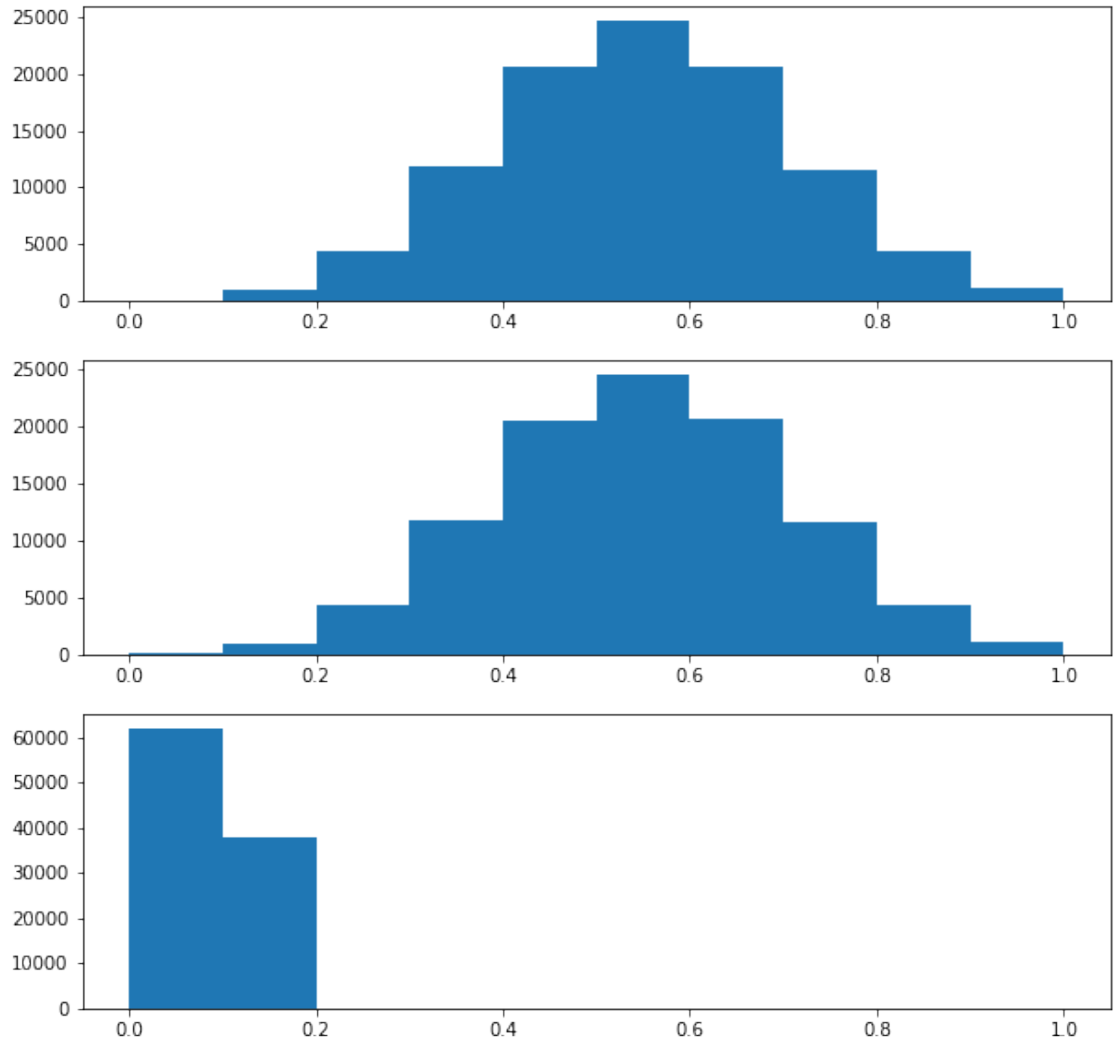
[66]:
```
# 2 b)

n_run = 100000
n_coin = 1000
n_flip = 10

V1_vec = np.zeros(n_run)
Vrand_vec = np.zeros(n_run)
Vmin_vec = np.zeros(n_run)

for run in tqdm(range(n_run)):
    run_i = np.zeros(n_coin)
    for flip in range(n_flip):
        experiment = np.random.randint(2, size=n_coin)
        run_i += experiment
```

```
    V_i = run_i/n_flip
    V1_vec[run] = V_i[0]
    Vmin_vec[run] = np.min(V_i)
    Vrand_vec[run] = np.random.choice(V_i,size=1)
```

```
  0%|          | 0/100000 [00:00<?, ?it/s]
```

[67]:
```
bins = np.arange(11)/10
plt.figure(figsize=(10,10))
plt.subplot(3,1,1)
plt.hist(V1_vec, bins=bins, density=False)
plt.subplot(3,1,2)
plt.hist(Vrand_vec, bins=bins, density=False)
plt.subplot(3,1,3)
plt.hist(Vmin_vec, bins=bins, density=False)
```

[67]:
```
(array([6.2204e+04, 3.7794e+04, 2.0000e+00, 0.0000e+00, 0.0000e+00,
        0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```

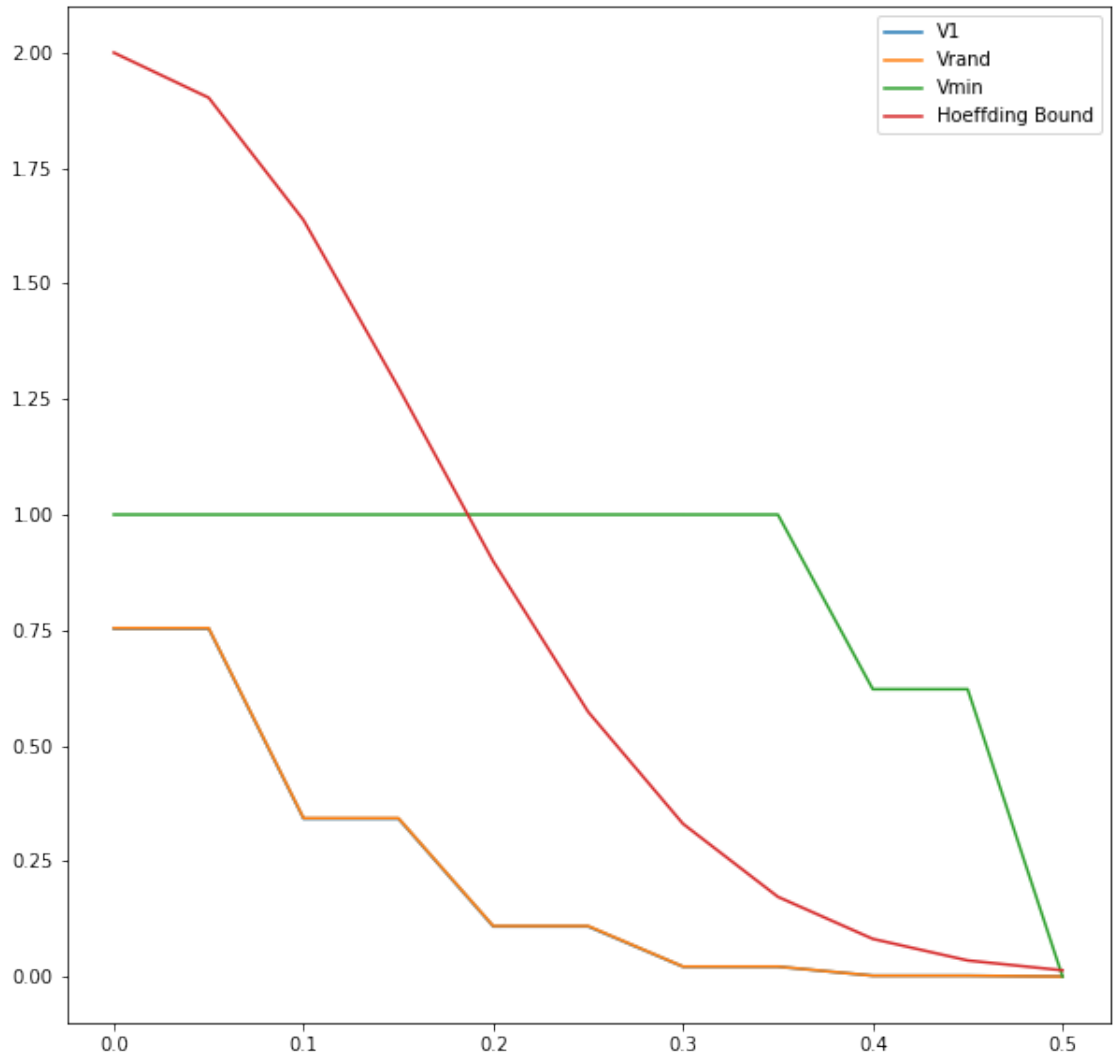


[74]:
```
epsilon_vec = np.linspace(0,0.5, 11, endpoint=True)
hoeffding_bound = 2*np.exp(-2*epsilon_vec**2*n_flip)
```

[90]:
```
mu = 0.5
P1_vec = np.zeros(epsilon_vec.shape)
Prand_vec = np.zeros(epsilon_vec.shape)
Pmin_vec = np.zeros(epsilon_vec.shape)

for i,epsilon in enumerate(epsilon_vec):
    P1_vec[i] = np.sum(np.abs(V1_vec-mu) > epsilon)/n_run
    Prand_vec[i] = np.sum(np.abs(Vrand_vec-mu) > epsilon)/n_run
    Pmin_vec[i] = np.sum(np.abs(Vmin_vec-mu) > epsilon)/n_run
```

[94]:
```
plt.figure(figsize=(10,10))
plt.plot(epsilon_vec,P1_vec, label='V1')
plt.plot(epsilon_vec,Prand_vec, label='Vrand')
plt.plot(epsilon_vec,Pmin_vec, label='Vmin')
plt.plot(epsilon_vec,hoeffding_bound, label='Hoeffding Bound')
plt.legend()
```

[94]: <matplotlib.legend.Legend at 0x7fea848c5760>



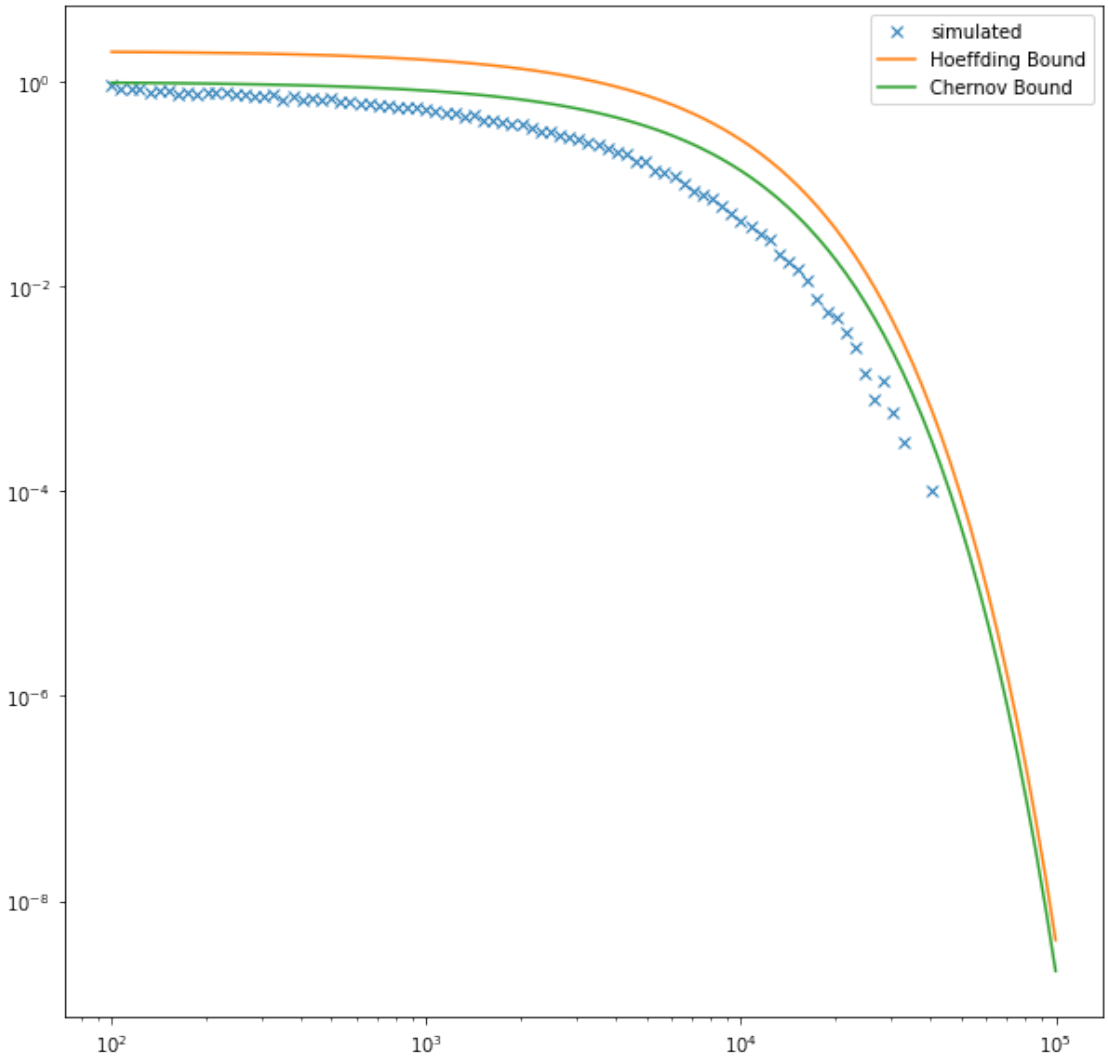
1.1 2 d)

Only $coin_{min}$ does not obey Hoeffding's Bound. $coin_1$ and $coin_{rand}$ are nearly identical and obey the bound. Because coin_min is not randomly chosen.

2 Exercise 3

[97]:
```
p = 0.5
epsilon = 0.01
Nset = np.round(np.logspace(2,5,100)).astype(int)
x = np.zeros((10000,Nset.size))
prob_simulate = np.zeros(100)
prob_hoeffding = np.zeros(100)
prob_chernov = np.zeros(100)

for i in range(Nset.size):
    N = Nset[i]
    x[:,i] = stats.binom.rvs(N, p, size=10000)/N
    prob_simulate[i] = np.mean((np.abs(x[:,i]-p)>epsilon).astype(float))
    prob_hoeffding[i] = 2*np.exp(-2*N*epsilon**2)
    beta = 1+(0.5+epsilon)*np.log2(0.5+epsilon) + (0.5-epsilon)*np.log2(0.
 ↪5-epsilon)
    prob_chernov[i] = 2**(-beta*N)

plt.figure(figsize=(10,10))
plt.loglog(Nset, prob_simulate, 'x', label='simulated')
plt.loglog(Nset, prob_hoeffding, label='Hoeffding Bound')
plt.loglog(Nset, prob_chernov, label='Chernov Bound')
plt.legend()
```

[97]:
```
<matplotlib.legend.Legend at 0x7fea84bf3fa0>
```


simulated
Hoeffding Bound
Chernov Bound
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{2}
10^{3}
10^{4}
10^{5}

[ ]: