HW3_Q2_Hardik_2678294168

September 24, 2021

# 1 EE660-HW3 : Hardik (2678294168)

[15]:
```
import numpy as np
from random import sample
```

[4]:
```
#generating input space of 1 and -1, with P[x=-1]=u
def generate_samples(input_space_size,u):
    num_neg=int(input_space_size*u)
    num_pos=input_space_size-num_neg
    input_space=[1]*num_pos
    for i in range(num_neg):
        input_space.append(-1)
    return input_space
```

[20]:
```
#function for drawing samples from input space
def draw_samples(input_space,size):
    dataset=sample(input_space,size)
    return dataset
```

[22]:
```
#function to calculate error rate in dataset
def error_rate(data):
    count_neg=0
    for i in data:
        if i==-1:
            count_neg=count_neg+1
    error=count_neg/len(data)
    return error
```

[23]:
```
#generating input space of 1000000 samples, with u=0.2
input_space1=generate_samples(1000000,0.2)
```

[35]:
```
dataset_10=draw_samples(input_space1,10)
```

[36]:
```
error_rate_10=error_rate(dataset_10)
print("E(h)_10 = {}".format(error_rate_10))
```

```
E(h)_10 = 0.3
```

[65]:

```
def experiment(u,n):
    input_space=generate_samples(1000000,u)
    error_rate_100times=[]
    for i in range(100):
        data_sample=draw_samples(input_space,n)
        error=error_rate(data_sample)
        error_rate_100times.append(error)

    print(" ")
    print("Q2_b_i : N={}, u={}".format(n,u))
    print(" ")
    print("Max E_d_{}(h) = {}".format(n,max(error_rate_100times)))
    print(" ")
    print("Min E_d_{}(h) = {}".format(n,min(error_rate_100times)))
    print(" ")
    print("Sample Mean E_d_{}(h) = {}".format(n,sum(error_rate_100times)/
 ↪len(error_rate_100times)))
    print(" ")
    print("Sample Standard Deviation E_d_{}(h) = {}".format(n,np.
 ↪std(error_rate_100times)))
    print(" ")

    count_err_less=0
    for err in error_rate_100times:
        if err!=u:
            count_err_less=count_err_less+1
    print("Q2_b_ii : N={}, u={}".format(n,u))
    print(" ")
    print("# of runs with error rate different than {} ={}".
 ↪format(u,count_err_less))
    print("")

    print("Q2_b_iii : N={}, u={}".format(n,u))
    print(" ")
    count_diff=0
    for err in error_rate_100times:
        if abs(err-u)<0.05:
            count_diff=count_diff+1
    print("P[|E_d_{}(h)-{}|<0.05] = {}".format(n,u,count_diff/100))

    count_learn=0
    if u==0.5:
        for err in error_rate_100times:
            if err<=0.45:
                count_learn=count_learn+1
        print("# of test datasets with error<=0.45 = {}".format(count_learn))
        print(" ")
```

```
```

## 2 Q2:b:

[78]:
```
experiment(0.2,10)
```

```
Q2_b_i : N=10, u=0.2

Max E_d_10(h) = 0.4

Min E_d_10(h) = 0.0

Sample Mean E_d_10(h) = 0.187

Sample Standard Deviation E_d_10(h) = 0.1188738827497445

Q2_b_ii : N=10, u=0.2

# of runs with error rate different than 0.2 =71

Q2_b_iii : N=10, u=0.2

P[|E_d_10(h)-0.2|<0.05] = 0.29
```

## 3 Q2:c:

[67]:
```
experiment(0.2,100)
```

```
Q2_b_i : N=100, u=0.2

Max E_d_100(h) = 0.32

Min E_d_100(h) = 0.1

Sample Mean E_d_100(h) = 0.20159999999999997

Sample Standard Deviation E_d_100(h) = 0.04187409700518926

Q2_b_ii : N=100, u=0.2

# of runs with error rate different than 0.2 =92

Q2_b_iii : N=100, u=0.2

P[|E_d_100(h)-0.2|<0.05] = 0.78
```

[68]:
```
experiment(0.5,10)
```

```
Q2_b_i : N=10, u=0.5

Max E_d_10(h) = 0.8

Min E_d_10(h) = 0.1

Sample Mean E_d_10(h) = 0.47400000000000003

Sample Standard Deviation E_d_10(h) = 0.15074481748969015

Q2_b_ii : N=10, u=0.5

# of runs with error rate different than 0.5 =70

Q2_b_iii : N=10, u=0.5

P[|E_d_10(h)-0.5|<0.05] = 0.3
# of test datasets with error<=0.45 = 40
```

[69]:
```
experiment(0.5,100)
```

```
Q2_b_i : N=100, u=0.5

Max E_d_100(h) = 0.61

Min E_d_100(h) = 0.34

Sample Mean E_d_100(h) = 0.4959999999999998

Sample Standard Deviation E_d_100(h) = 0.05033885179461287

Q2_b_ii : N=100, u=0.5

# of runs with error rate different than 0.5 =89

Q2_b_iii : N=100, u=0.5

P[|E_d_100(h)-0.5|<0.05] = 0.69
# of test datasets with error<=0.45 = 22
```