6.Experiments with distance concentration

(a)

Step1: sample n=100 points for d=2,5,10,20,100 at random from S^{d-1}: draw standard normal distribution $Z_1, Z_2 \dots Z_d$ and then compute $X = \frac{Z}{||Z||}$

Step2: compute $\binom{n}{2}$ distance from these points

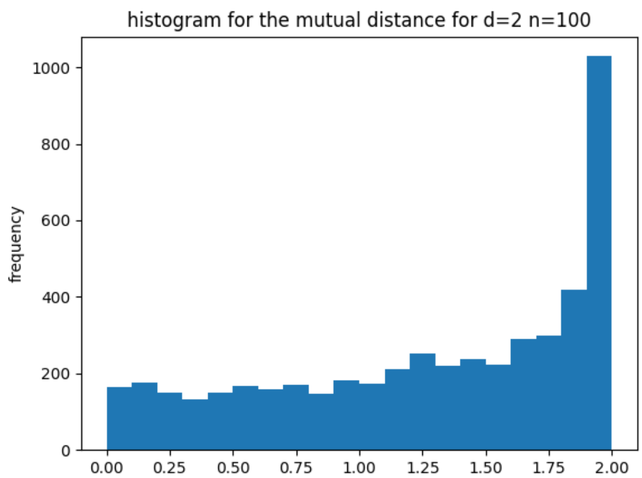
Step3: make a histogram (0-2, 20 bins)

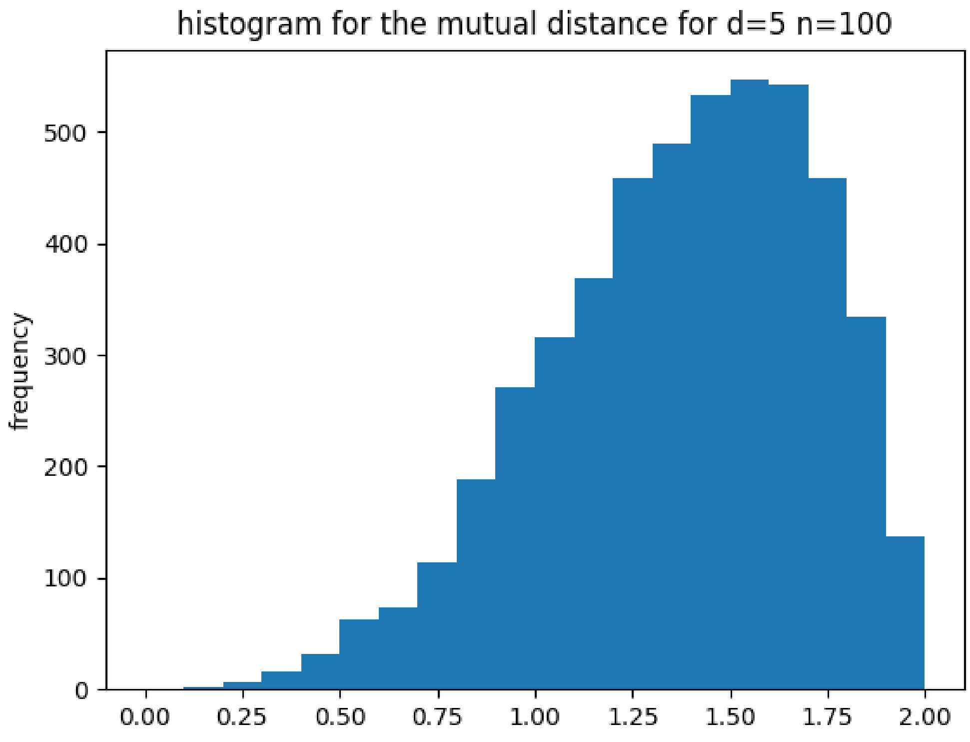
The program code:

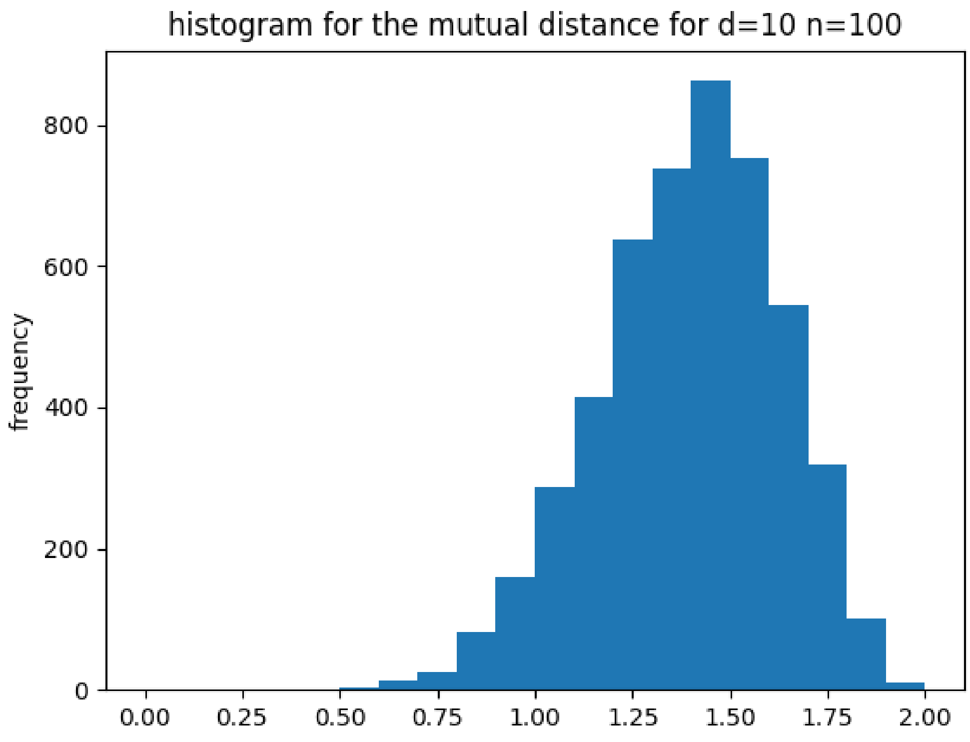
```
import math
import matplotlib.pyplot as plt
import numpy as np

def hist(x,d):
    title="histogram for the mutual distance for d="+str(d)+" n=100"
    plt.title(title)
    plt.ylabel("distance")
    plt.ylabel("frequency")
    plt.hist(x,bins=20,range=[0,2])
    plt.show()

def sph_sample(d,n):
    """
    generate n random points from d-dimensional sphere
    :param d:
    :param n:
    :return:
    """
    X_list=[]
    Z_list=[]
    for i in range(n):
        # normal dis
        Z=[]
        znorm=0
        for j in range(d):
            value=np.random.normal()
            Z.append(value)
```

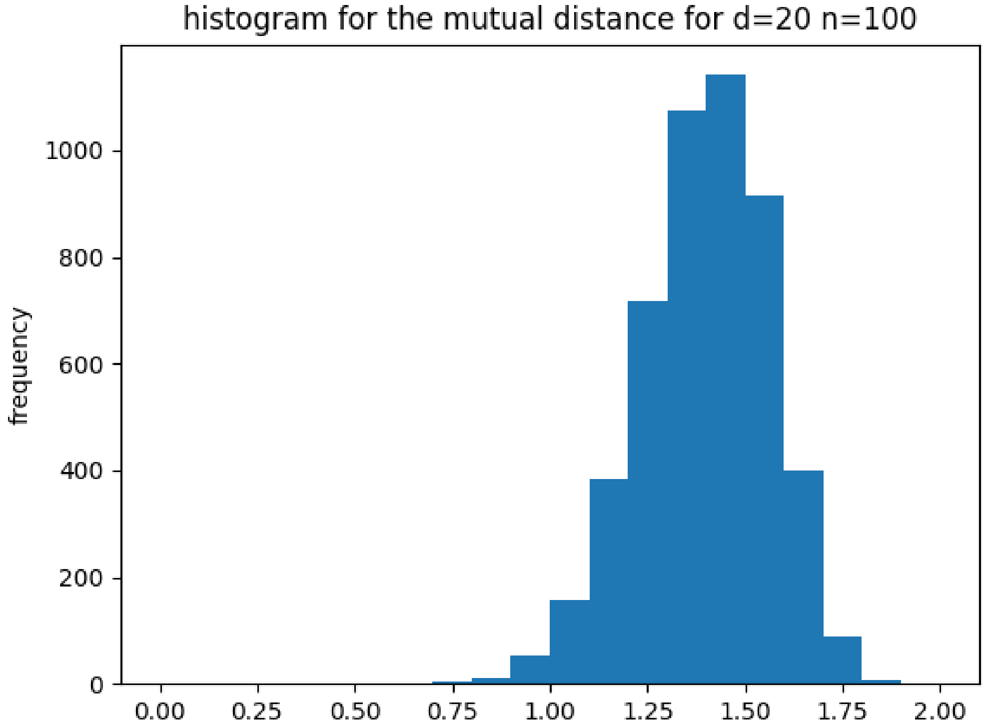
```
            znorm+=value**2
        # normalize to sphere
        znorm=math.sqrt(znorm)
        X=[]
        for j in range(d):
            X.append(Z[j]/znorm) #
        X_list.append(X)

    return X_list

def dis_mutual(x):
    dislist=[]
    n=len(x)
    d=len(x[0])
    for i in range(n-1):
        a=x[i]
        for j in range(i+1,n):
            b=x[j]
            dissqr=0
            for k in range(d):
                dissqr += (a[k]-b[k])**2
            dislist.append(math.sqrt(dissqr))
    return dislist

d=[2,5,10,20,100]
for i in range(5):
    x=sph_sample(d[i],100)
    dislist=dis_mutual(x)
    hist(dislist,d[i])
```

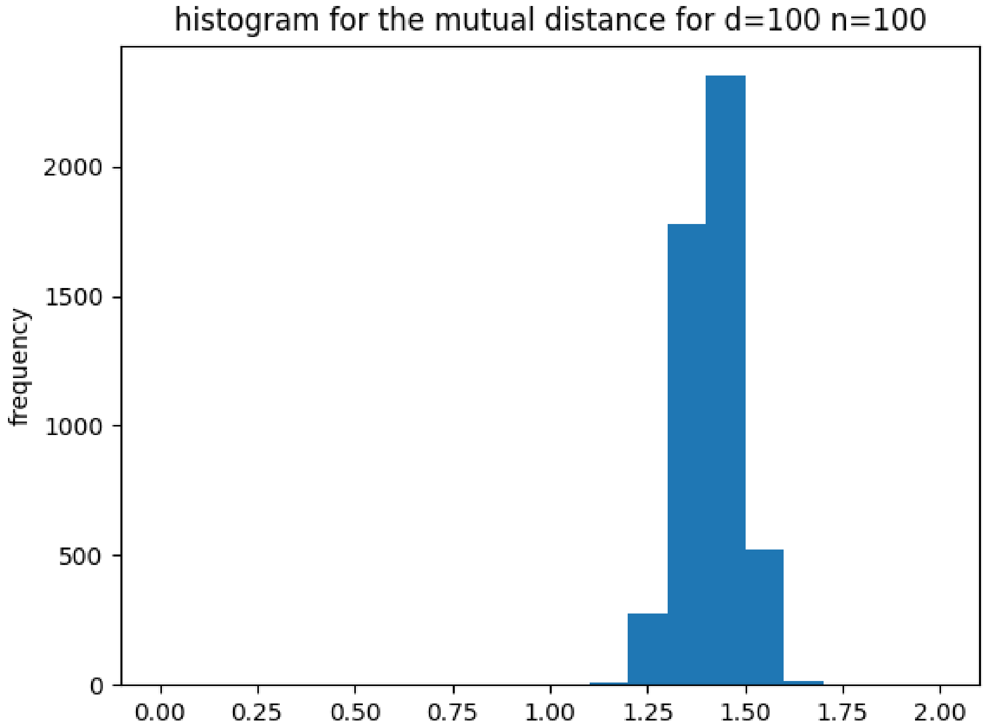
the result:




histogram for the mutual distance for d=5 n=100
frequency
0
100
200
300
400
500
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


histogram for the mutual distance for d=10 n=100
frequency
0
200
400
600
800
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


histogram for the mutual distance for d=20 n=100
frequency
0
200
400
600
800
1000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


histogram for the mutual distance for d=100 n=100
frequency
0
500
1000
1500
2000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

(b)

According to the trend in the histogram graph in (a), as d increase, the distance distributed mainly in [1.30,1.40], it almost around 1.43

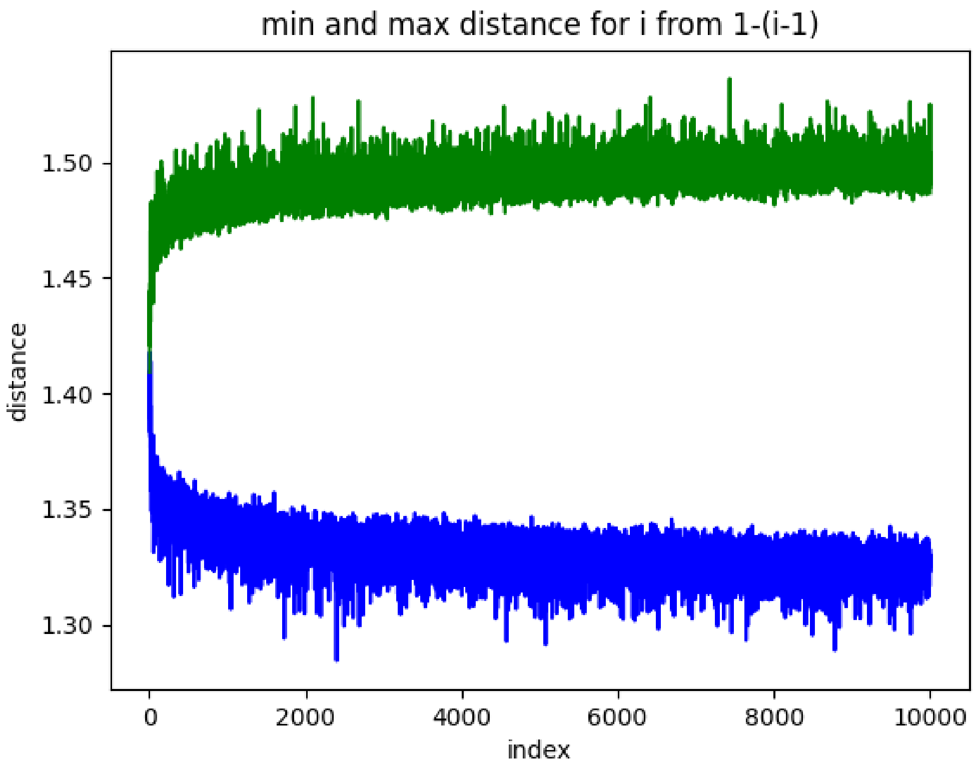
(c)

Here is the programing code:

```
import math
import matplotlib.pyplot as plt
import numpy as np

def sph_sample(d,n):
    """
    generate n random points from d-dimensional sphere
    :param d:
    :param n:
    :return:
    """
    X_list=[]
    Z_list=[]
    for i in range(n):
        # normal dis
        Z=[]
        znorm=0
        for j in range(d):
            value=np.random.normal()
            Z.append(value)
            znorm+=value**2
        # normalize to sphere
        znorm=math.sqrt(znorm)
        X=[]
        for j in range(d):
            X.append(Z[j]/znorm) #
        X_list.append(X)

    return X_list

```

```python
def dis_euclidsqr(a,b,n):
    dissqr=0
    for k in range(n):
        dissqr += (a[k] - b[k]) ** 2
    # return math.sqrt(dissqr)
    return dissqr

def dis_minmax(x,d):
    minlist=[]
    maxlist=[]
    n=len(x)
    for i in range(1,n):
        print(i)
        mind=4
        maxd=0
        for j in range(i):
            dis=dis_euclidsqr(x[i],x[j],d)
            if mind>dis:
                mind=dis
            if maxd<dis:
                maxd=dis
            # mind=min(mind,dis)
            # maxd=max(maxd,dis)
        minlist.append(math.sqrt(mind))
        maxlist.append(math.sqrt(maxd))
    return [minlist,maxlist]

def fvalplt(y):
    x=np.arange(2, len(y[0]) + 2)
    plt.title("min and max distance for i from 1-(i-1)")
    plt.xlabel("index")
    plt.ylabel("distance")
    colval = ['b', 'g', 'r', 'c', 'm', 'y', 'k', 'w']
    for i in range(len(y)):
        plt.plot(x, y[i], color=colval[i])
    plt.show()

d=1000
x=sph_sample(d,10000) # it seems that only d is large enough, it will
concentrate
minmaxlist=dis_minmax(x,d)
fvalplt(minmaxlist)
```

here is the result plot



From the plot we can see that as the i increases, the min and max tend to stabilize which means the distance of all these 10000 points will eventually locate in a narrow range, [1.30,1.50]. That is to say there is a high possibility that there are lots of points with approximately the same distance from each other.