CS425

Assignment I- Path Loss Exponent

Course Instructor :

Dr. Amitangshu Pal

Submitted By :

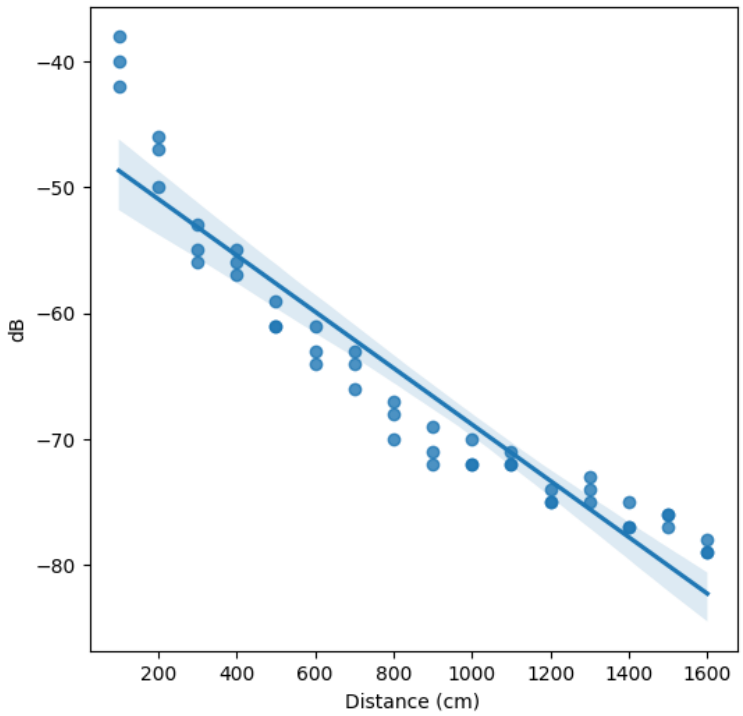
Nishant Roshan
200643

Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur
January 29, 2023

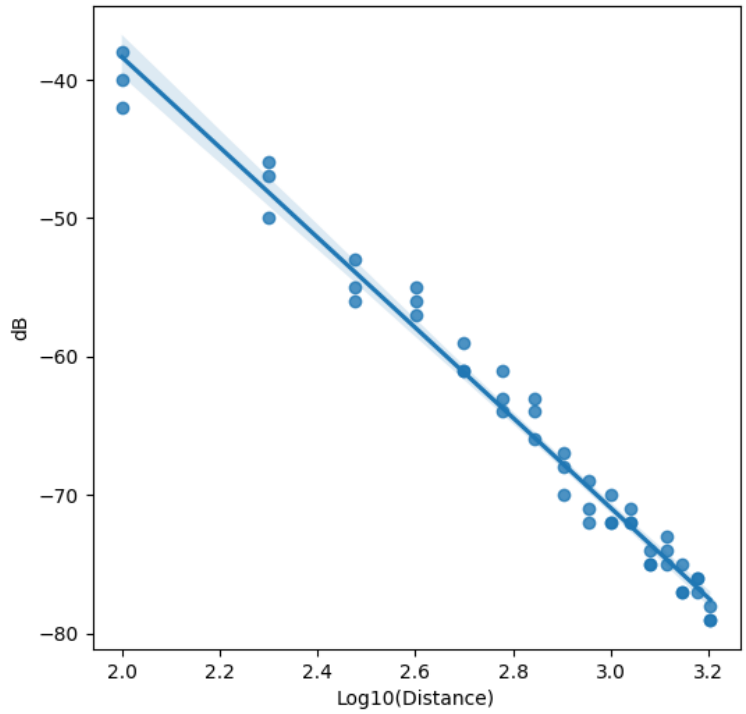
0.1 Estimating the path loss exponent

- The data was collected on Hall 3 terrace, IITK. The purpose of choosing the location was to ensure that there is least reflection, scattering or attenuation due to nearby objects.
- The devices used were two mobile phones, on one of which hotspot was activated, while the other was connected to it. The intention was to choose an almost isotropic point source.
- I used the wifi-analyser app to measure the signal strength at different positions.
- For any particular distance from the source, the data was collected for three different orientations of the receiver device.
- The CSV file with recorded data and the jupyter notebook containing code to evaluate the data is provided in the links.

The following curve is obtained for Signal Strength (dBm) vs Distance (cm):



On replacing Distance with log of Distance, we obtain the following plot:



I found the best-fit line for the collected data points using the *Regplot*() library of seaborn, which returned the line which deviates the least from the collected set of points, by minimising mean squared loss. The line returned was

$$y = mx + c,\ slope(m) = -32.5413,\ intercept(c) = 26.6953$$

This equation corresponds to the formula

$$P_r(d) = -10nlog_{10}(\frac{d}{d_0}) + P_r(d_0) \implies path\ loss\ exponent = \frac{abs(m)}{10} = 3.254$$

Moreover to check how well the best fit line estimates the actual values, I calculated the R2 squared norm . The value close to 1 indicates that the fit is good. The R2 squared norm for my sample comes out approximately 0.98, which implies a fairly good estimate.
The code for the same is:

```
slope, intercept = np.polyfit(df['distance'], df['decibel'], 1)
print("Slope :",slope,"\nIntercept :",intercept)
print("Path loss exponent : ",abs(slope/10))
print("The r2 norm is",r2_score(df['decibel'],slope*df['distance']+intercept))
```

```
[4]  ✓ 0.1s                                                    Python
Slope : -32.5412990352722
Intercept : 26.69532299989519
Path loss exponent :  3.2541299035272204
The r2 norm is 0.9813018268949326
```

The variance of the RSSI samples w.r.t. best fit line came out to be 2.224 This variance is due to external attenuations like people, walls, and other noise. The code for calculating variance is:

```
x = np.array(df['distance'])
y = np.array(df['decibel'])

a, b = np.polyfit(x, y, 1)

z = []
sum = 0
for i in range(1,len(x)):
    z.append(a*x[i-1]+b - y[i-1])

print("Variance is ",np.var(z))
```

[5] ✓ 0.1s Python

```
Variance is  2.224855733881004
```

0.2 Problem 2- Estimating Range Error

$$P_r(d) = -10nlog_{10}(\frac{d}{d_0}) + P_r(d_0) \implies d = d_0 * 10^{\frac{P_r(d_0) - P_r(d)}{10n}} \quad (1)$$

- I started by setting the reference distance(d_0) as 100cm and measured the signal strength there. Lets us call it $P_r(d_0)$. On measurement, we get

$$P_r(d_0) = -39.756dBm.$$

- Then I randomly chose five different points and measured the signal strength there. Let's call these $P_r(d)$.
- This I used the value of path loss exponent calculated in the previous part, which is 3.254
- So we ultimately have all the variables on the RHS of equation (1). Hence we can calculate the expected d values. Compare these values with the actual distance between the transmitter and receiver to get the error.

```
n= 3.2541
P_d_0 = -39.756

readings = { 4:-59, 7:-66, 10: -72 ,13: -76, 16: -79.33}
error= 0.0
for dist in readings:
  readings[dist] =  np.round(pow(10,(P_d_0 - readings[dist])/(10*n)), decimals=5)
  print('Estimated distance calculated from given strength at actual distance ', dist, 'metres is', readings[dist], 'metres.\nError at actual distan
  error += abs(readings[dist]-dist)

error/=5
print('Avg error over 5 samples in the distance calculation is:', np.round(error, decimals=5), 'metres')
```

[6] ✓ 0.2s Python

```
Estimated distance calculated from given strength at actual distance  4 metres is 3.90281 metres.
Error at actual distance  4 metres is 0.09719
Estimated distance calculated from given strength at actual distance  7 metres is 6.40457 metres.
Error at actual distance  7 metres is 0.59543
Estimated distance calculated from given strength at actual distance  10 metres is 9.79204 metres.
Error at actual distance  10 metres is 0.20796
Estimated distance calculated from given strength at actual distance  13 metres is 12.99556 metres.
Error at actual distance  13 metres is 0.00444
Estimated distance calculated from given strength at actual distance  16 metres is 16.44854 metres.
Error at actual distance  16 metres is 0.44854
Avg error over 5 samples in the distance calculation is: 0.27071 metres
```

From the above result we can now get the absolute error which is given as: 0.27 meters
Error is a result of the following factors:
1) Scattering/dispersion : Due to obstacles in the way while measuring the signal.
2) Human error: There must be some error in measuring distance between the transmitter and receiver.
3) Other sources like random error, mobile application error etc.

The error range above (approx 27cm) is much smaller compared to distance of order 10 metres and above and thus the dataset we collected and the calculations performed are considerable.

References :
1) CS425A Lecture Notes (Semester 22-23-II)
2) https://pythonprogramming.net/how-to-program-best-fit-line-machine-learning-tutorial/

Thank You