CS425: COMPUTER NETWORKS

Assignment 1

Danish Mehmood
Roll Number: 210297

February 4, 2024

Path Loss Exponent

Distance (m.)	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5	Average RSSI
1	24	23	18	20	22	21.4
3.58	37	30	35	35	34	34.2
5.33	39	38	37	39	38	38.2
7.45	47	40	44	42	42	43
9.3	45	47	40	47	46	45

Table 1: RSSI measurements at different distances

- All the distances are in meters and readings in dBm.
- The signs of the readings have been flipped since they all were negative.
- Calculation for the best line of fit was done using `Python` libraries `Numpy` and `SciPy`. The function `stats.linregress` from the `stats` module of `SciPy` was also used.
- The best fit line was found out by performing linear regression on the data collected.
- The code used is as follows:

```python
import numpy as np
from scipy import stats

data = [
    (1, 24), (1, 23), (1, 18), (1, 20), (1, 22),
    (3.58, 37), (3.58, 30), (3.58, 35), (3.58, 35), (3.58, 34),
    (5.33, 39), (5.33, 38), (5.33, 37), (5.33, 39), (5.33, 38),
    (7.45, 47), (7.45, 40), (7.45, 44), (7.45, 42), (7.45, 42),
    (9.3, 45), (9.3, 47), (9.3, 40), (9.3, 47), (9.3, 46)
]

x, y = zip(*data)

x = np.array(x)
y = np.array(y)

# Taking the log of x values
log_x = np.log(x)

# Performing linear regression
slope, intercept, r_value, p_value, std_err = stats.linregress(log_x, y)

print(f"Slope: {slope}, Intercept: {intercept}")
```

Listing 1: Python code for finding best fit line

- The slope (m) and intercept (c) obtained as a result of the above code are:

$$m \approx 20.61 \quad c \approx 21.10$$

- Hence the best fit line of the data is:

$$y = 20.61x + 21.1$$

where x is the log of distance from the WiFi AP and y is negative of RSSI.

- The plot of RSSI values (in dBm) v/s log of distances (in meters) was plotted using the `Python` library `matplotlib`. Some values were generated from the best fit line to get a smooth straight plot. The code is as follows:

```python
import matplotlib.pyplot as plt

# Generating x values for the best-fit line
log_x_fit = np.linspace(min(log_x), max(log_x), 100)
x_fit = np.exp(log_x_fit)  # Convert back to the original scale using exp
y_fit = slope * log_x_fit + intercept  # Calculate y values for the line

# Plotting the original data points
plt.scatter(x, y, color='blue', label='Data points')

# Plotting the best-fit line
plt.plot(x_fit, y_fit, 'r-', label='Best-fit line')

# Setting the x and y-axis labels
plt.xlabel('x (log scale)')
plt.ylabel('y')
plt.xscale('log')  # Set x-axis to log scale to match the transformation

# Adding a legend
plt.legend()

plt.show()
```

Listing 2: Python code for plotting the best fit line

- The plot of best fit line obtained is as follows:

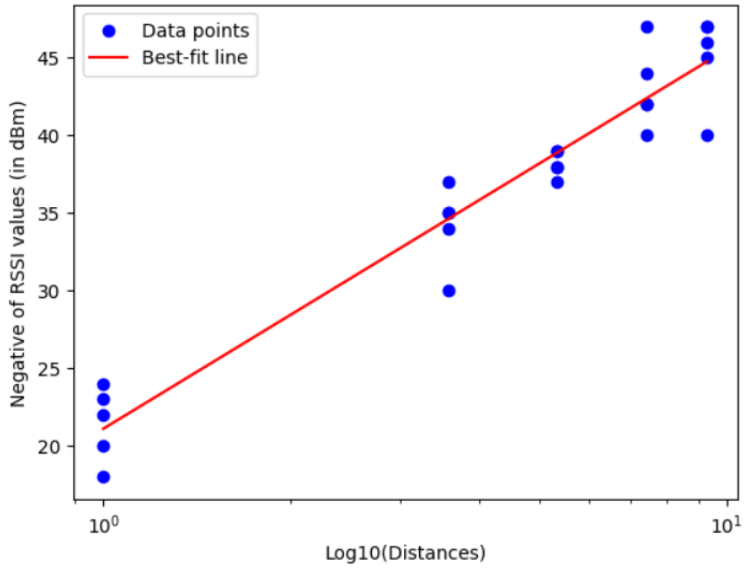

Figure 1: Plot of best fitting line of RSSI values (in dBm) v/s log of distances (in meters).

- The path loss exponent n can be found by dividing the slope m by 10, i.e.,

$$n = \frac{m}{10} \approx 2.061$$

  We have divided m by 10 and not -10 as the negative values were already flipped in the readings and hence the equation of the line.

- Now to find variance σ^2 with respect to line of best fit we have used a Python program as follows:

```python
import numpy as np

data = [
    (1, 24), (1, 23), (1, 18), (1, 20), (1, 22),
    (3.58, 37), (3.58, 30), (3.58, 35), (3.58, 35), (3.58, 34),
    (5.33, 39), (5.33, 38), (5.33, 37), (5.33, 39), (5.33, 38),
    (7.45, 47), (7.45, 40), (7.45, 44), (7.45, 42), (7.45, 42),
    (9.3, 45), (9.3, 47), (9.3, 40), (9.3, 47), (9.3, 46)
]

x, y = zip(*data)

x = np.array(x)
y = np.array(y)

# Take the log of x values
log_x = np.log(x)

# Given line equation: y = 20.61x + 21.10
```

```python
# Calculating predicted y values
y_pred = 20.61 * log_x + 21.10

# Calculating residuals (differences between actual and predicted y values)
residuals = y - y_pred

# Calculating the variance of the residuals
variance = np.var(residuals)

print(f"Variance of the 'y' values with respect to the line: {variance}")
```

Listing 3: Python code for finding variance

- The variance from the above code comes out to be

$$\sigma^2 \approx 4.81$$

- So the best fit line, the path loss exponent n and the variance σ^2 respectively are:

$$y = 20.61x + 21.1 \quad m \approx 20.61 \quad n \approx 2.061$$

Range Estimation

- The $P_r(d_0)$ from the best fit line is **-21.1 dBm**.
- But after taking some readings at a distance of 1m the average value of $P_r(d_0)$ came out to be **-23.2 dBm**.
- Then readings were taken for the following distances:

Actual Distance	Reading 1	Reading 2	Reading 3	Reading 4	Reading 5	Average Reading	Calculated Distance	Error
5.3	40	42	40	37	41	40	6.3	1.0
8.3	42	44	40	47	44	43.5	9.3	1.0
10.2	45	49	43	47	46	46	12.2	2.0
10.5	45	47	42	47	43	44.8	10.7	0.2
12.5	51	46	43	48	47	47	13.6	1.1

Table 2: RSSI readings and estimated ranges

- All distances are in meters and RSSI values in dBm with flipped signs.
- Average error is **1.06 meters**.
- Python was used for the calculations and the formula used for calculating distance is:

$$d = d_0 \cdot 10^{(P_r - 23.2)/10n} = 10^{(P_r - 23.2)/10n}$$

where P_r is the RSSI reading with flipped sign, n is the path loss exponent and d_0 is 1 meter.