EEE4463 Artificial Intelligence and Big Data

Lab 7 – Data Analytic and Modeling

**Name: \_\_\_\_Yeung Wing\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Class: \_\_EG114403-1B\_\_\_\_\_\_\_\_\_**

**Objective**: To analyze data, to condition data, to build model based on conditioned data and to extract information.

**Tools**: Windows PC

**Software**: Anaconda3 version 2021.11 individual edition

Spyder version 5.1.5

**Topics covered**:

* Analyze and condition data using linear regression method and RANSAC regression method
* Build model using conditioned data
* Extract information from model parameters

**Component list**:

Line.csv

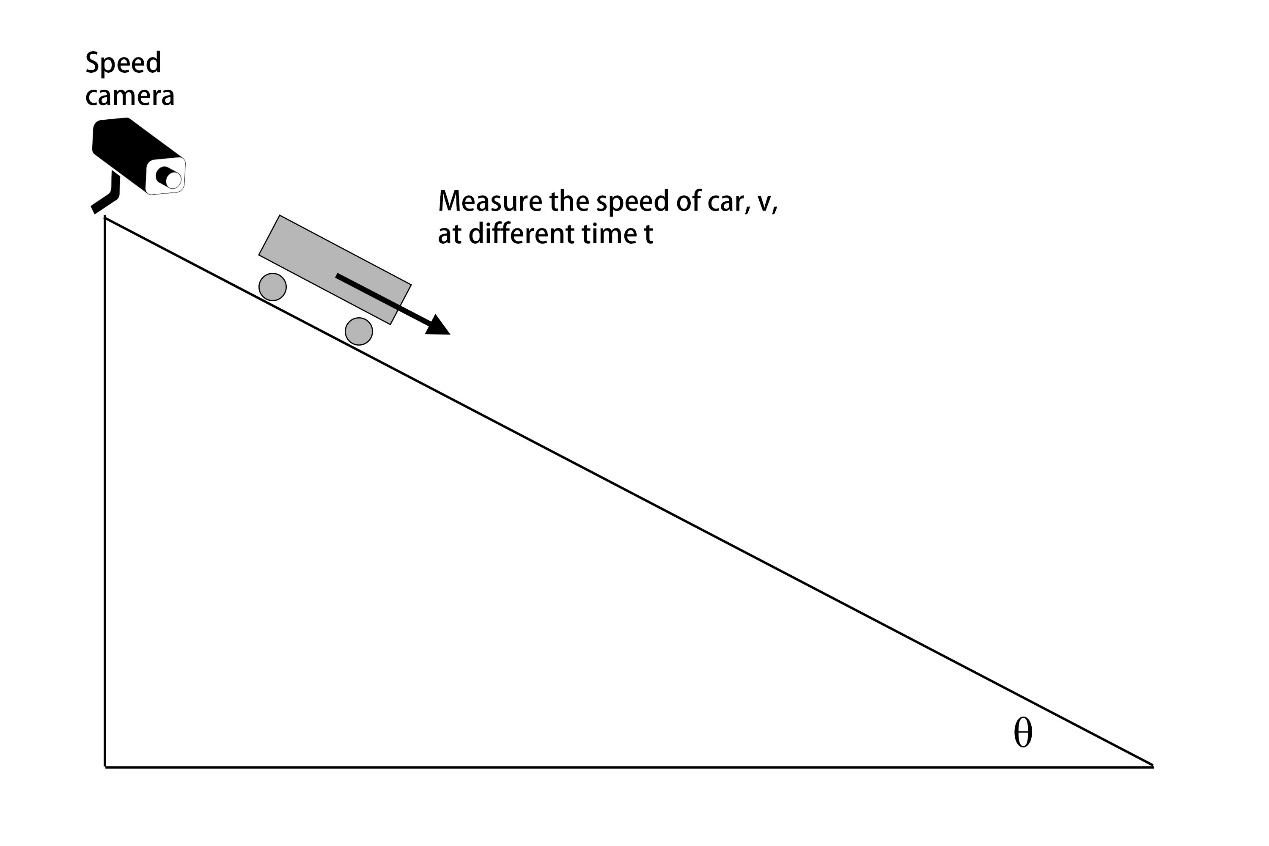
LinearRegression08.py

RANSAC\_RegressorLineExample08.py

# Introduction

In this laboratory exercise, you are going to analyze some data using linear regression and RANSAC methods. Before building a model to describe data, it is necessary to condition the data so that outliers are removed. In our case, our data may be corrupted with noise and outliers. In this lab, we are going to study the use of linear regression and RANSAC methods to analyze the data under the influence of noise and outliers. After filtering the outliers, we build a model to model the conditioned data and extract insight from the model built.

We have a practical scenario that a car is placed on a ramp. The car moves down the slope due to gravity. We have a speed camera to measure the speed of the car at different time t. The measured speed in m/s and the time t in second are saved in a csv file named “Line.csv”. We assume the data stored in this file is free of noise and outliers. Our task is to study the use of linear regression method and RANSAC regression method to analyze data under the influence of noise and outliers. After filtering outliers, a model is built based on the conditioned data. Finally, we extract as much insight as possible from the model built.



## Part A: Ideal case – data with no noise and outliers

The csv file “Line.csv” stores speed and time data of the car moving down the slope. The data essentially has no noise and outliers. It is the ideal situation that the speed camera working perfectly measuring the speed of the car, v, at different time, t. A Python program with filename “LinearRegression09.py” is given. Place the “Line.csv” file in the same folder as “LinearRegression08.py” file and run the program.

**Question 1.1:** **Capture and save the figures generated by the program.**

|  |
| --- |
|  |

**Question 1.2: Briefly describe what is the program doing. What can you tell from the figures?**

|  |
| --- |
| 1. The program reads the csv file and converts it to a numpy array. 2. After that, it uses matplotlib to plot a scatterplot of the linear regression with speed and time. 3. Then it uses the LinearRegression.fit from sklearn's linear\_model module to print the coefficients and intercept. 4. After plotting a line of the LR model on the previous figure. 5. It builds the second figure which is a residual histogram. 6. Finally, it prints the prediction in the case where the time (x) is 14s, the speed of the car (y) is around 83.6 m/s.   From the figure, we can see that the LR model is completely on top of the data, because the data provided is already noiseless. This makes the residual histogram useless at this point. |

**Question 1.3: Comment if the linear regression method can accurately model data.**

|  |
| --- |
| The linear regression can accurately model the data and predict the data.  > Linear regression model (without noise): Coefficients = [[4.9]] intercept = [15.]  > y = [4.9] \* x + [15.]  > When x = [14.], y = [83.6] |

## Part 2: Data corrupted with noise

In this part, we add some Gaussian noise to the speed data to simulate the speed signal being corrupted with noise. The following line of Python code generates noise and adds to vector y.

|  |
| --- |
| #add some noise to y  y = y + np.random.normal(0, 1.5, size=(n\_samples, 1)) |

**Question 2.1: Briefly describe the meaning of the above line of code.**

|  |
| --- |
| Draw random samples from a normal (Gaussian) distribution.  np.random.normal(loc, scale, size)  > loc: Mean of the distribution.  > scale: spread of the distribution  > size: Output shape  Which mean we add some gaussian noise to y by creating a [n\_samples x 1] array containing 0 with 1.5 spread of the distribution and add it to y  y is speed; n\_samples is the length of time array |

**Task:**

1. Save the file “LinearRegression08.py” as a new filename “LinearRegression\_withNoise.py”.
2. Add the above lines of code to the program in appropriate location.
3. Change the lines of code as follows.

|  |
| --- |
| plt.scatter(  x, y, color="orange", marker=".", label="data **with noise**"  ) |

|  |
| --- |
| plt.title('Linear regression - data **with noise**') |

|  |
| --- |
| print( f'Linear regression model (**with noise**): Coefficients = {reg.coef\_} intercept = {reg.intercept\_}' ) |

|  |
| --- |
| plt.title('Residual histogram (data **with noise**)') |

1. Run the program “LinearRegression\_withNoise.py”

**Question 2.2: Capture and save the figures generated by the program.**

|  |
| --- |
|  |

**Question 2.3: What is the estimated model? What can you tell from the figures? Does the noise have large influence to the estimated model?**

|  |
| --- |
| > Linear regression model (with noise): Coefficients = [[4.89138762]] intercept = [15.02910932]  > y = [4.89138762] \* x + [15.02910932]  > When x = [14.], y = [83.50853595] |

**Question 2.4: Comment the appropriateness in using linear regression method to model data which is under the influence of noise.**

|  |
| --- |
|  |

## Part 3: Data corrupted with noise and outliers

In this part, in addition to adding noise to the data, we add some outliers to test the linear regression method. We will add the following lines of codes to the program in part 2.

|  |
| --- |
| # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # Create outlier data  n\_outliers = 200  np.random.seed(0)  x\_outlier = np.random.normal(8.0, 1.0, size=(n\_outliers, 1))  y\_outlier = np.random.normal(10.0, 0.5, size=(n\_outliers, 1))  # Plot the outliers  plt.scatter(  x\_outlier, y\_outlier, color="green", marker=".", label="outliers"  )  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  # append outliers to x and y  x = np.concatenate( (x, x\_outlier), axis=0)  y = np.concatenate( (y.reshape(n\_samples, 1), y\_outlier), axis=0 )  # \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

**Question 3.1: Briefly describe the meaning of the above lines of code.**

|  |
| --- |
|  |

**Task**

1. Save the file “LinearRegression\_withNoise.py” in part 2 as a new filename “LinearRegression\_withNoiseAndOutlier.py”.
2. Add the above lines of code to the program in appropriate location.
3. Change the following lines of code as follows.

|  |
| --- |
| #add some noise to y  y = y + np.random.normal(0, 1.5, size=(n\_samples, 1))  plt.scatter(  x, y, color="orange", marker=".", label="data **with noise and outliers**"  ) |

|  |
| --- |
| plt.title('Linear regression - data **with noise and outliers**') |

|  |
| --- |
| print( f'Linear regression model (with **noise and outliers**): Coefficients = {reg.coef\_} intercept = {reg.intercept\_}' ) |

|  |
| --- |
| plt.title('Residual histogram (data **with noise and outliers**)') |

1. Run the program “LinearRegression\_withNoiseAndOutlier.py”

**Question 3.2: Capture and save the figures generated by the program.**

|  |
| --- |
|  |

**Question 3.3: What is the estimated model? What can you tell from the figures? Does the outliers have large influence to the estimated model?**

|  |
| --- |
|  |

**Question 3.4: Comment the appropriateness in using linear regression method to model data which is under the influence of noise and outliers.**

|  |
| --- |
|  |

## Part 4: Use of RANSAC regression method

In part 3, we note that linear regression method is not a good method to model data under the influence of noise and outliers. To solve this problem, we use RANSAC regression method instead. Provided program file “RANSAC\_RegressorLineExample08.py” is a Python program making use of RANSAC regression method to model the data corrupted with noise and outliers.

**Task**

1. Copy the file “RANSAC\_RegressorLineExample08.py” in the same folder having the file “Line.csv”.
2. Run the program.

**Question 4.1: Capture and save the figures generated by the program.**

|  |
| --- |
|  |

**Question 4.2: Briefly describe the meaning of the codes in “RANSAC\_RegressorLineExample08.py”.**

|  |
| --- |
|  |

**Question 4.3: What is the final estimated model? What can you tell from the figures? Does the outliers have large influence to the estimated model?**

|  |
| --- |
|  |

**Question 4.4: Comment the appropriateness in using RANSAC regression method to model data which is under the influence of noise and outliers.**

|  |
| --- |
|  |

## Part 5: Extract information from the estimated model

In this part, we are going to extract information from the estimated model. It is known to us that the relationship between speed and time in this case is

 (Equation 1)

Where

v is the speed (velocity) at time t

u is the initial speed at time t = 0

g is the gravitational acceleration (9.8)

t is the time

 is the angle of the ramp

**Question 5.1: Extract as much information using the Equation 1 and the estimated line equation in part 4.**

|  |
| --- |
|  |

**Question 5.2 (challenging question): Can you think of other application making use of the experiment setup and method?**

|  |
| --- |
|  |

## Part 6 (Challenging): Prove the Equation 1 in part 5

**Question 6.1: Make use of your knowledge of Physics, prove that equation 1 in part 5 is valid (correct).**

|  |
| --- |
|  |

<end>