“[ENPM 673] HOMEWORK - 1”

ENPM 673 – PERCEPTION FOR AUTONOMOUS ROBOTS

HW1 REPORT

ADITYA VAISHAMPAYAN -116077354

Instructor: Dr. Cornelia Fermuller

INDEX:

Contents

Abstract:

The purpose of this homework was to understand the principles of linear regression. The task was to fit a line to a two-dimensional data using different linear least square techniques. It also gave us knowledge about the eigen values and the eigen vectors and how they affect the spread of the data. This homework also gave us a chance to implement a model that is robust to the affect of the noise and thus result in a best line fit for the given data.

Q1) EIGEN VALUES AND EIGEN VECTORS

My understanding of eigen vectors and eigen values:

We obtain the eigen vectors and the eigen values do determine the shape of the data. Eigen values generally represent the covariance of the data whereas the eigenvectors represent the linear dependence between the directions of the spread of data. Thus, by examining the covariance matrix we can infer the variance and covariances of the data. The variance represents how well the data is spread and the covariance represents the orientation of the data.

If we would like to represent the covariance matrix with a vector and its magnitude, we should simply try to find the vector that points into the direction of the largest spread of the data, and whose magnitude equals the spread (variance) in this direction.

In other words, the largest eigenvector of the covariance matrix always points into the direction of the largest variance of the data, and the magnitude of this vector equals the corresponding eigenvalue. The second largest eigenvector is always orthogonal to the largest eigenvector, and points into the direction of the second largest spread of the data.

Many problems present themselves in terms of an eigenvalue problem:

$$A.v = \lambda.v$$

In this equation **A** is an n-by-n matrix, **v** is a non-zero n-by-1 vector and λ is a scalar (which may be either real or complex). Any value of λ for which this equation has a solution is known as an eigenvalue of the matrix **A**. It is sometimes also called the characteristic value. The vector, **v**, which corresponds to this value is called an eigenvector. The eigenvalue problem can be rewritten as

$$A.v - \lambda.v = 0$$

$$A.v - \lambda.I.v = 0$$

$$(A - \lambda.I).v = 0$$

If **v** is non-zero, this equation will only have a solution if:

$$|A - \lambda.I| = 0$$

This equation is called the characteristic equation of **A** and is an n^{th} order polynomial in λ with n roots. These roots are called the eigenvalues of **A**.

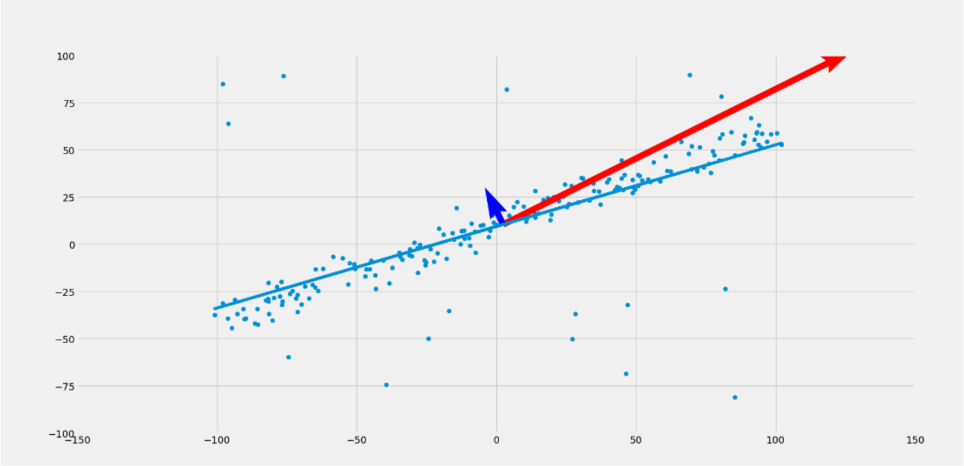

Fig: Eigen vectors for data1

DATA SET 1:

- Eigen values for data 1: $[4138.24045932 \quad 438.60191761]$
- Eigen vectors for data 1: $\begin{bmatrix} 0.89578578 & -0.44448604 \\ 0.44448604 & 0.89578578 \end{bmatrix}$
- Covariance matrix for data 1: $\begin{bmatrix} 3407.3108669 & 1473.06388943 \\ 1473.06388943 & 1169.53151003 \end{bmatrix}$

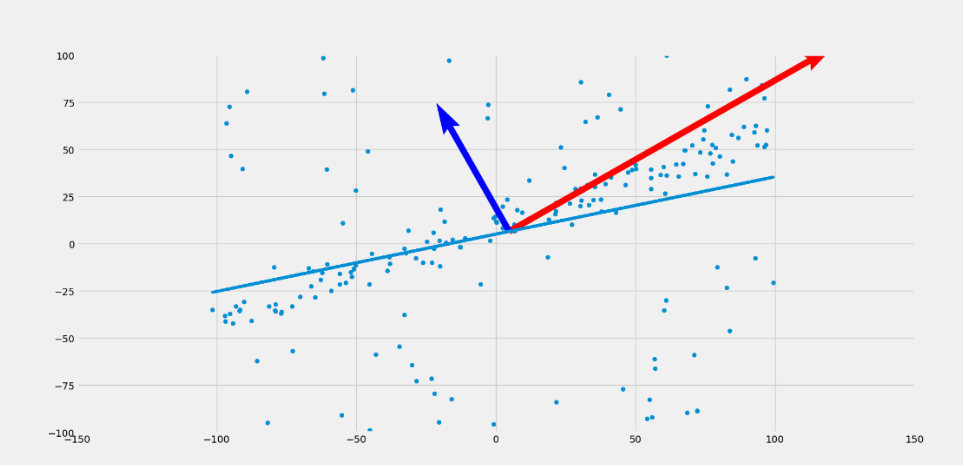

Fig: Eigen vectors for data2

DATA SET 2:

- Eigen values for data 2: $[4008.7296 \quad 1587.287]$
- Eigen vectors for data 2: $\begin{bmatrix} 0.8695 & -0.4937 \\ 0.4937 & 0.8695 \end{bmatrix}$
- Covariance matrix for data 2: $\begin{bmatrix} 3418.2909 & 1039.7574 \\ 1039.7574 & 2177.7259 \end{bmatrix}$

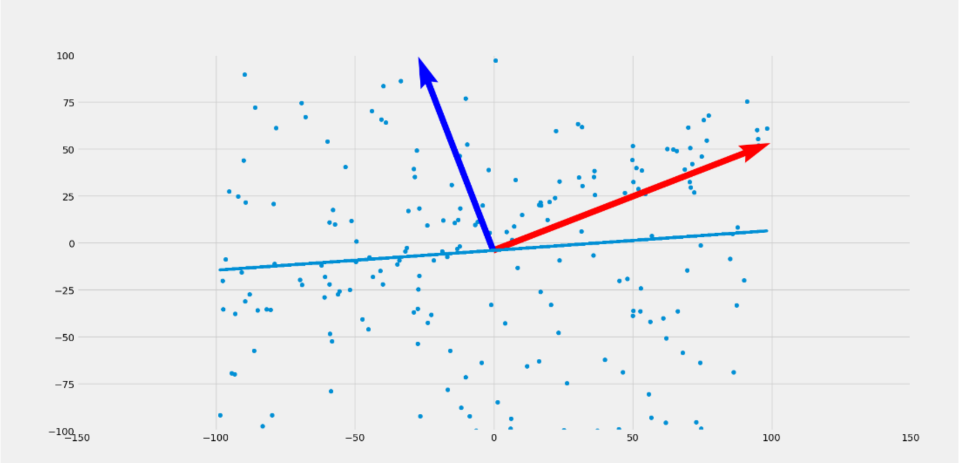

Fig: Eigen vectors for data3

DATA SET 3:

- Eigen values for data 3: $[3212.2725 \quad 2243.2197]$
- Eigen vectors for data 3: $\begin{bmatrix} 0.9326 & -0.36077 \\ 0.36077 & 0.93265 \end{bmatrix}$
- Covariance matrix for data 3: $\begin{bmatrix} 3086.1413 & 326.065 \\ 326.0654 & 2369.3509 \end{bmatrix}$

Q2) LINE FITTING USING LEAST SQUARES

Least squares is one of the methods to find the best fit line for a dataset using linear regression. The most common application is to create a straight line that minimizes the sum of squares of the errors generated from the differences in the observed value and the value anticipated from the model. Least-squares problems fall into two categories: linear and nonlinear squares, depending on whether or not the residuals are linear in all unknowns.

Steps in calculating the least squares:

1. Calculate the mean of x values and the mean of y values
2. Calculate the slope of the best fit line using the following formula
3. Calculate the y-intercept of the line by the below formula
4. The best fit line is called the regression line and it has the least square of distance from each data point to the line

Calculating the least squares can be done in two ways:

- Ordinary Least Squares (**sum of vertical distances from the line is minimized**)
- Total Least Squares (**sum of perpendicular distances from the line is minimized**)

1) ORDINARY LEAST SQUARES

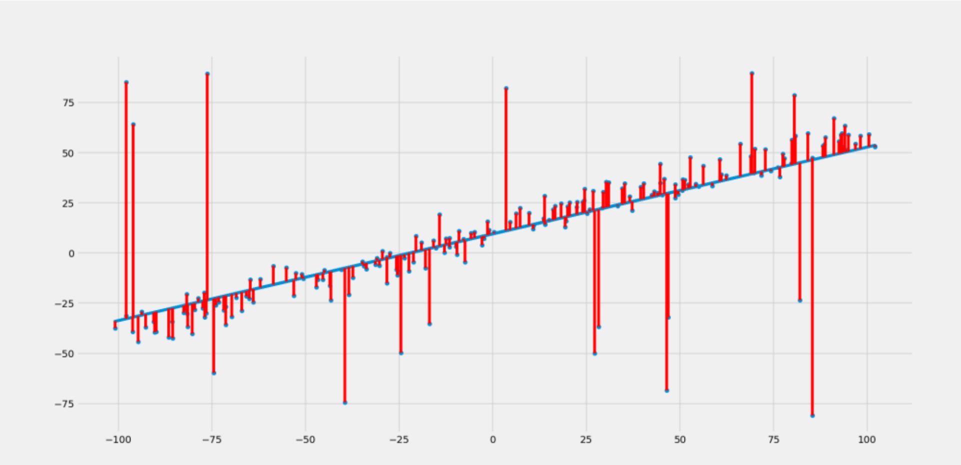

Fig: Line fitting on data1 using Ordinary Least Squares (OLS)

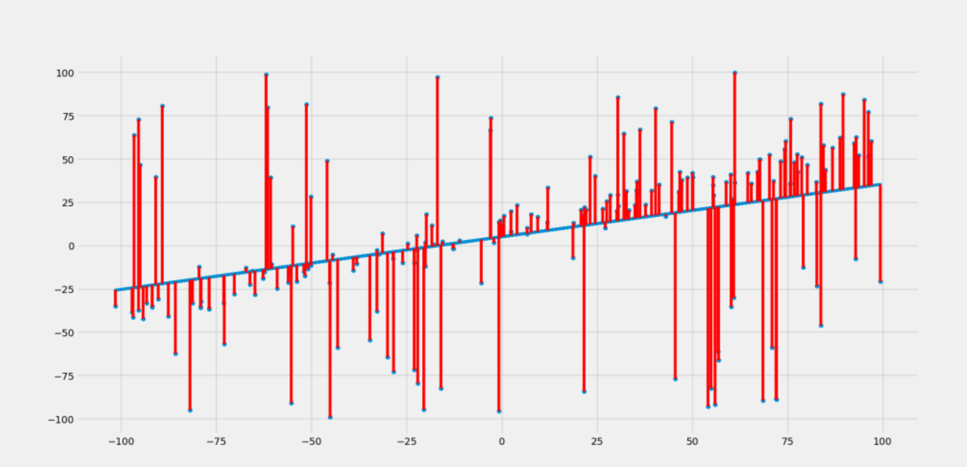

Fig: Line fitting on data using Ordinary Least Squares (OLS)

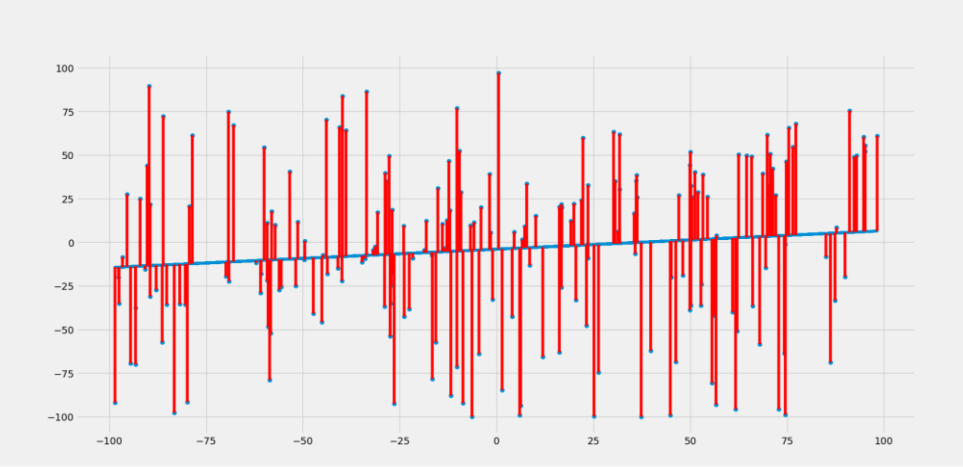

Fig: Line fitting on data 3 using Ordinary Least Squares (OLS)

2) TOTAL / ORTHOGONAL LEAST SQUARE vs VERTICAL LEAST SQUARE

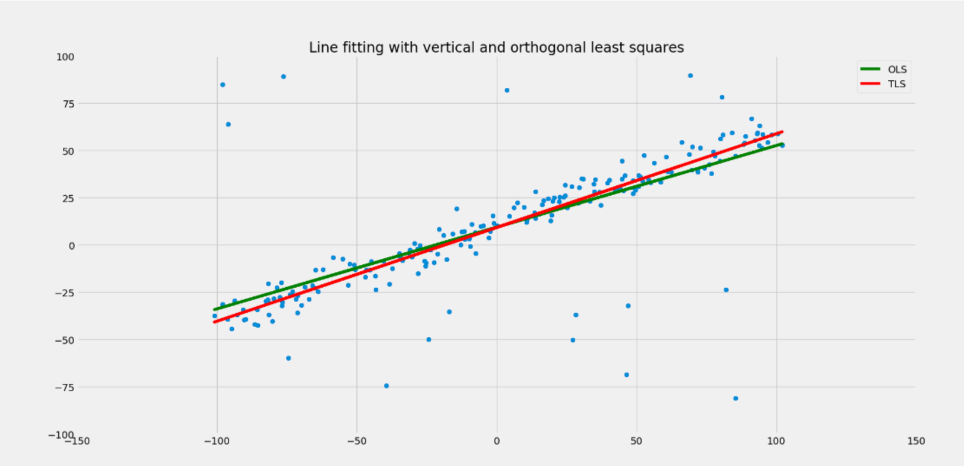

Fig: Line fitting using OLS and TLS on data 1

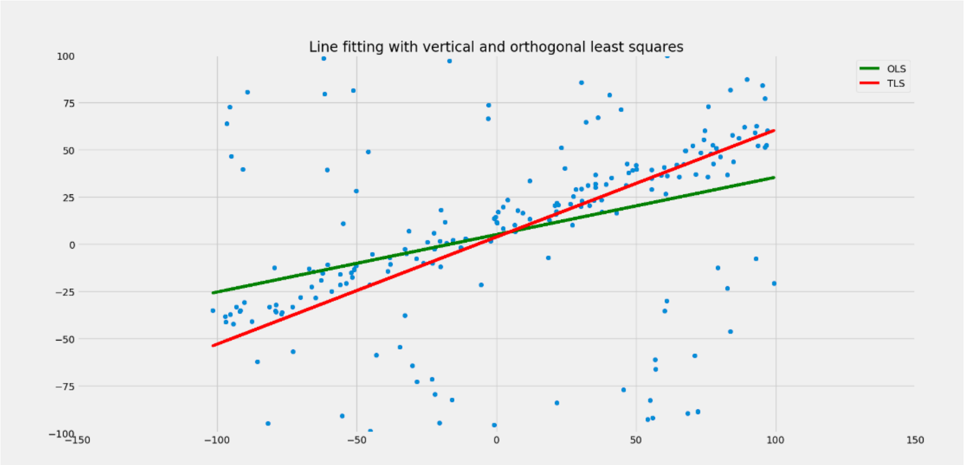

Fig: Line fitting using OLS and TLS on data 2

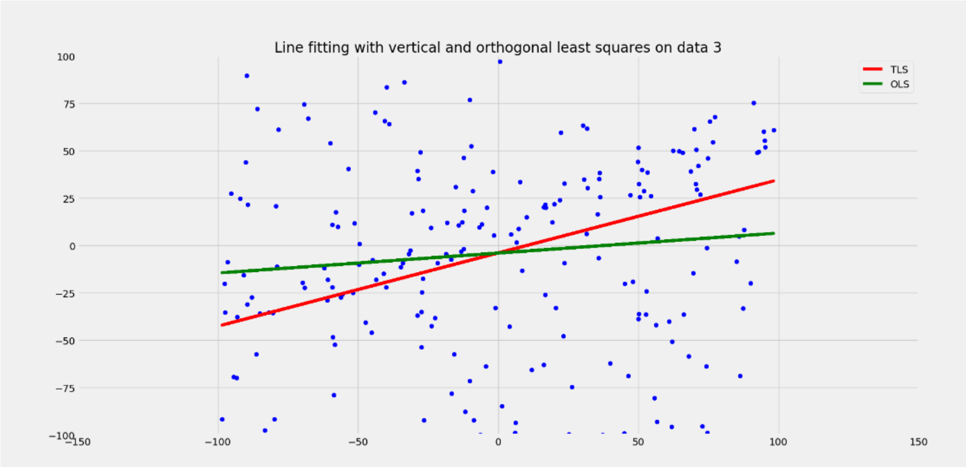

Fig: Line fitting using OLS and TLS on data 3

Q3) OUTLIER REJECTION USING RANSAC

RANSAC stands for random sample consensus. The data generally gets corrupted with noise and in order to obtain a best fit line for the data we need to eliminate the noise. As seen in the Q2 where we do not reject the noise, our line doesn't fit the data robustly. It gets affected by the noise data points. The points that perfectly fit within the defined threshold are called the inliers whereas those that do not fit within the defined threshold are called outliers.

Algorithm for RANSAC:

1. Randomly select two data points from the sample
2. Consider those two points as inliers and draw a line between them.
3. Also, obtain the slope and intercept for that line and save it.
4. Count the distance of the remaining points from these two points and only choose those points whose distance is lesser than the defined threshold. The selected points are called inliers whereas the remaining points are considered as outliers
5. Run steps 2 - 4 for multiple iterations and if the program encounters an iteration where the number of inliers is greater than previous maximum counted inliers, update the previously obtained slope and intercept with the slope and intercept of the iteration with maximum no. of inliers.
6. Once the number of iterations are completed, the final value of the slope and intercept obtained is considered to be the parameters for the best fit line of the model

In the figures below I have shown the line fitting output after and before applying the RANSAC algorithm. The red line shows the line fit without the RANSAC algorithm. We can see that the red line is affected by the outlier points. And the green line shows the best fit line after applying the RANSAC algorithm. As we can see in the figures below that it is not getting affected by the presence of the outliers

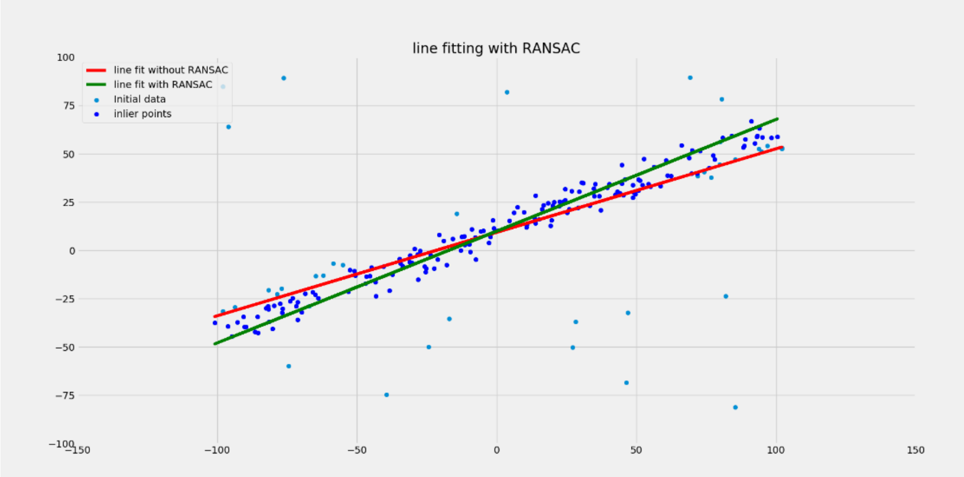

Fig: DATA 1 after applying RANSAC outlier rejection technique

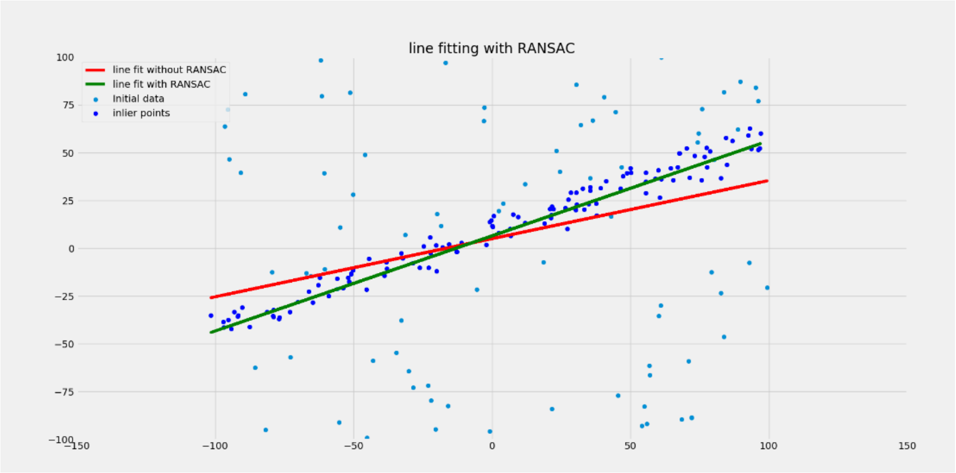

Fig: DATA 2 after applying RANSAC outlier rejection technique

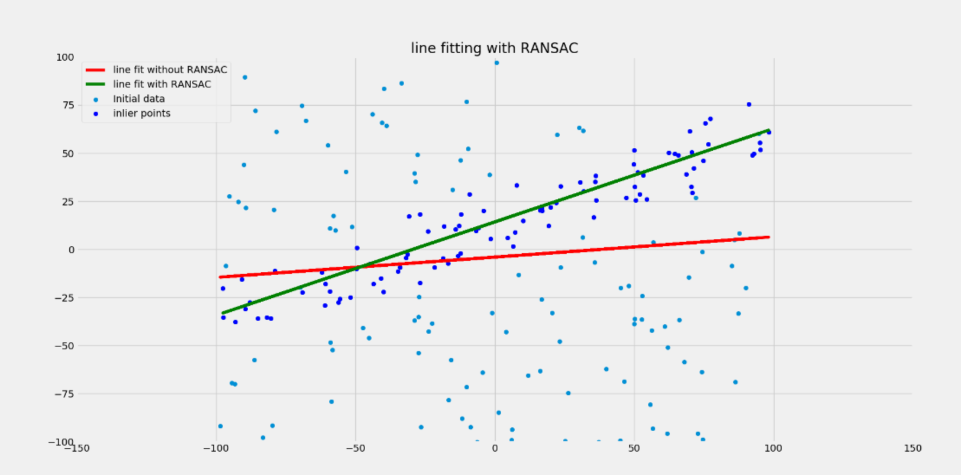

Fig: DATA 3 after applying RANSAC outlier rejection technique

Q4) REGULARIZATION

$$Error_{L2}(\beta) \;=\; \frac{1}{2}\sum_{i=1}^{n}(f(x_i;\,\beta) \,-\, y_i)^2 \,+\, \frac{L}{2}\sum_{k=1}^{n}(\beta_i)^2$$

Here L is lambda also called the regularization parameter. The above function is called ridge regression.

We write the relationship between the X and Y coordinates as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \dots & x_{k1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{12} & \dots & x_{k2} \\ 1 & x_{1n} & \dots & x_{kn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Here beta are the estimation parameters i.e. the slope and the intercept. We obtain β values as follows:

$$\beta \;=\; ((X^T X \;+\; L * I)^{-1} X^T)Y$$

In the figures below I haven’t considered the higher degree terms while implementing the Linearization algorithm because the NumPy arrays were getting stacked horizontally and it kept giving dimensionality error. Given below is the python file line using which I have implemented the above given equation

np.matmul(np.matmul(np.linalg.inv(np.matmul(matrix_x.T,matrix_x) + LI),matrix_x.T),Y)

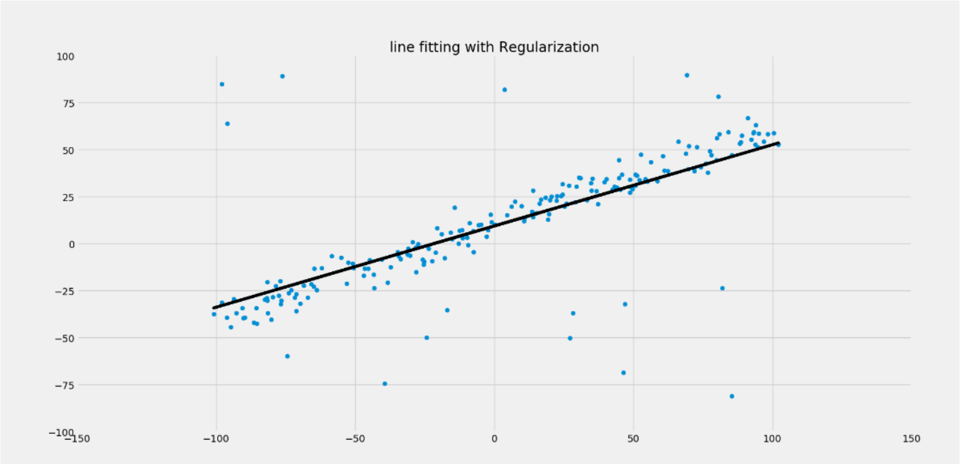

Fig: Line fitting with regularization on Data 1

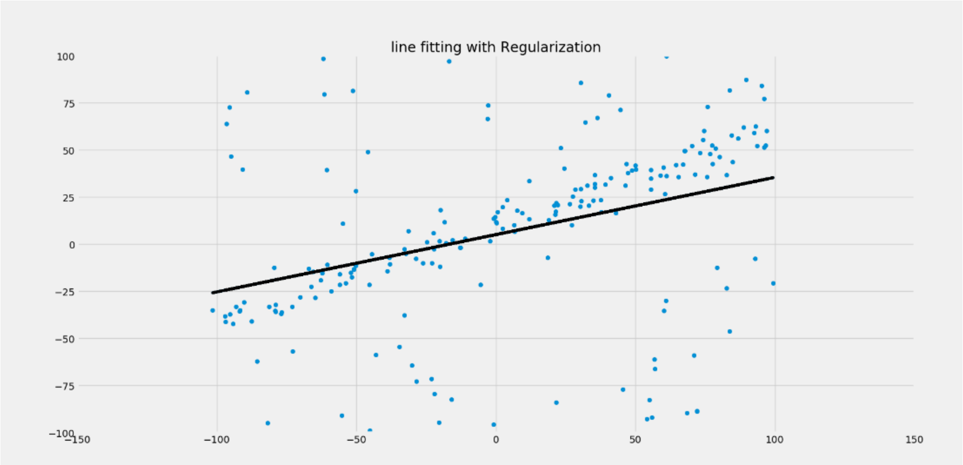

Fig: Line fitting with regularization on Data 2

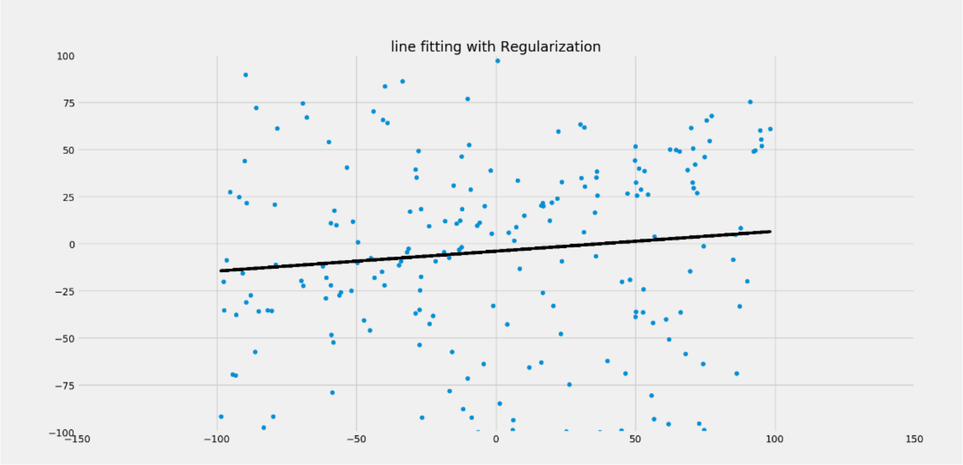

Fig: Line fitting with regularization on Data 3

My choice for outlier rejection technique:

1) DATA SET 1: **I prefer RASANC for dataset 1** as it is more robust to outlier rejection. However, **Ordinary least Squares and Total least Squares technique are as robust as RANSAC for this dataset** because in data set 1 the points are very close together and aren't affected by the noise that much. Hence the probability of choosing two random points and them being the inliers in RANSAC is equivalent to the optimization obtained by OLS and TLS. If we use RANSAC, then we will obtain a perfect fit in the initial few iterations itself.

2) DATA SET 2: Even though the data is skewed a little because of the presence of higher number of outliers, we can see that no of data points skewed by noise is lesser. **For this dataset we can either use Total Least Squares or RANSAC.** Just the probability of choosing two random points and them being the inliers in RANSAC is higher because the data is skewed way lesser as compared to dataset 3. Thus, we can prefer RANSAC or TLS algorithms to get an optimized line fit.

3) DATA SET 3: We can see that the data is highly skewed here. The number of points affected by the noise is large. So, **we can say that RANSAC is a good algorithm for this dataset** however it

isn’t the best one. The number of points affected by the noise is large. By using the RANSAC algorithm we can get a better output. But, we need to increase the no. of iterations while implementing the algorithm and also, we need to increase the threshold value so that higher no. of data points is considered. Since the data is really spread the chances of selecting two points from the good data points is lesser and hence, we increase the no. of iterations.

However, if we try the algorithm for ridge regression that is shown in the section of Regularization, there are chances that we might obtain a higher degree model that shall non-nearly fit the data. If we implement ridge regression, then we can also capture the non - linearities in the data perfectly. By tuning the parameter lambda, we can make sure that the ridge regression model does not over fit the data.

CONCLUSION

I was able to perform linear regression and obtain a model for best fitting the data using several techniques such as Linear Least Square optimization and RANSAC. I was also able to select which method to apply for a given data so that it gave optimized results.