ENPM 690 and HW Assignment 2
Robot Learning

Sumedh R Koppula

Febraury 14, 2022

1 Problem 1

1.1 Field of View of the camera

Given a camera with a resolution of 5MP with a square shaped sensor with a width and height of 14mm and focal length of the camera is 25mm

$$FOV = 2 * arctan(\frac{sensorsize}{2f}) \tag{1}$$

$$FOV = 2 * arctan(\frac{14}{50}) \tag{2}$$

$$\textbf{FOV = 0.55 rad} \tag{3}$$

1.2 Minimum number of Pixels

$$NumberofPixels = imagearea * (\frac{cameraresolution}{sensorarea}) \tag{4}$$

$$ImageSize = focallength * (\frac{objectsize}{objectdistance}) \tag{5}$$

sensor area = 14 * 14 = 196 mm^2
image area = image size * image size
sensor area = 0.0625 * 0.0625 = 0.00391 mm^2
Number of Pixels = 0.00391 * (5000000/196) = 99.65 pixels

2 Problem 2

2.1 Curve Fitting: Standard Least Sqaures for given videos

For curve fitting using Standard least square, Firstly, I have used Open CV inbuilt to read both the videos using cv2.videoCapture() and read() methods. I have stored all the image frames. Then, I have passed all the image frames into a method called ballCordinates to calcuate the top most and bottom most colored pixel.

Functionality of ballCordinates method:

Initially, Each frame is converted from BGR to HSV using cvtColor. The converted image is masked for lower and upper red ranges.The combined mask is fed to bitwiseand along with image frame to remove all non-red regions. Later, Each individaul red valued top and bottom pixels appended and returned.

Later, The obtained X and Y ball coordinates are stacked as an input matrix to leastSqaureCalculator. Inorder to calculate standard least square, A quadratic equation $y = ax^2 + bx + c$ is used. The array data is converted into a matrix A. Least square is calculated by substituting Matrix A and A_T into below equation:

$$A.[(A_T.A)^{-1}.(A_T.Y)] \tag{6}$$

The returned values are plotted and curve fitted as follows in Figure 1:

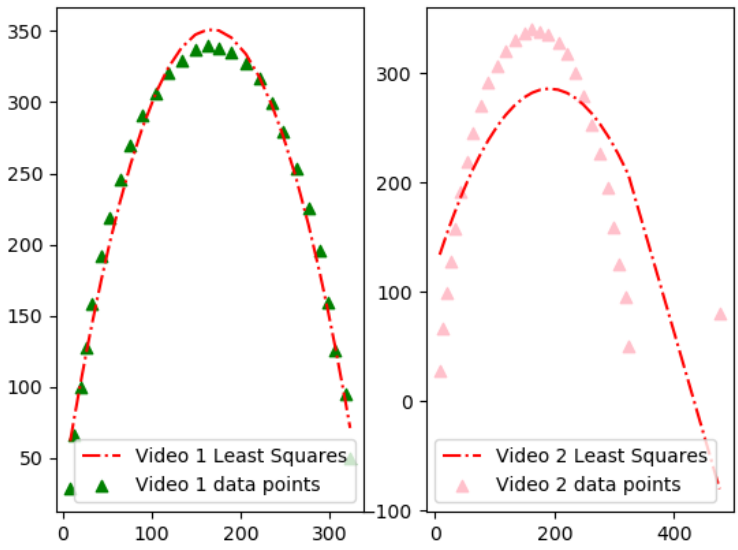

Figure 1: Curve Fitting using Standard Least Sqaures

3 Problem 3

3.1 Covariance Matrix with its eigen values and eigen vectors

Initially, From the given .CSV file using pandas I have retreived the data to store it in individual arrays as ages and insurance cost. The obtained arrays are stacked into data matrix. This matrix is then passed covariance matrix method to calculate covariance.

$$Covariance(x, y) = \sum_{0}^{n-1} \frac{(x - \bar{x})(y - \bar{y})}{n - 1} \tag{7}$$

Where:
X_i : the values of the X variable
Y_j : the values of the Y variable
$\bar{X}$: the mean (average) of the X variable
$\bar{Y}$: the mean (average) of the Y variable
n :the number of data points

$$\beta = (X^T X)^{-1} X^T \tag{8}$$

eigen values and vectors are calculated for the obtained matrix. Then these eigen vectors are then plotted as below:

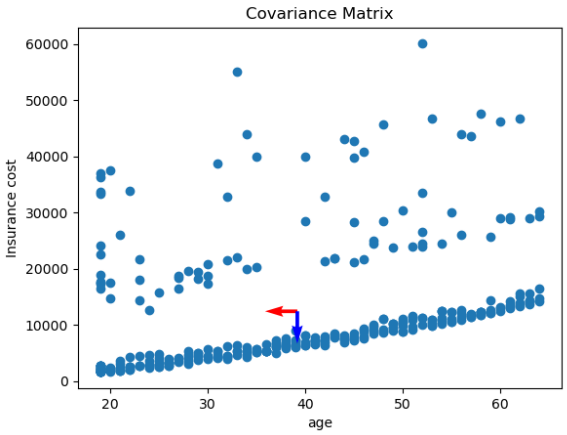

Figure 2:

3.2 Curve fitting using linear sqaure method, total square method and RANSAC

3.2.1 linear square method

The Least Square Method (LSM) is a mathematical approach for determining the best fit curve to a set of data points by minimizing the sum of the squares of residuals.

The discrepancy between the observed and estimated values of a dependent variable is known as residual.

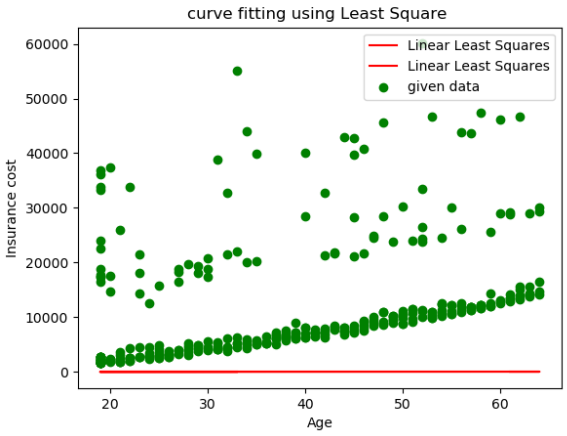

Figure 3: Curve Fitting using Least Square Method

From the defination, Normal equations for Least Square solution to XB = Y is given by

$$X_T X B = X_T Y \tag{9}$$

$$B = (X^T X)^{-1}(X^T Y) \tag{10}$$

Using the above equation, I had calculated Least square for line equation y = mx+c

3.2.2 Total square method

For total square method, the best fit line must pass through the center of mass of the set of points, center the data (move mass center to origin). As this reduces the problem of finding 2 parameters (the unit vector normal to the line). Then we can solve for d.
Inorder to best fit data using total sqaure method initially we calculate $U^T U$ and use eigenvalue decomposition

$$U = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \vdots & \vdots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \quad U^T U = \begin{bmatrix} \sum_{i=1}^{n}(x_i - \bar{x})^2 & \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \\ \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) & \sum_{i=1}^{n}(y_i - \bar{y})^2 \end{bmatrix}$$

The line equation for curve fitting is calculated by finding the smallest eigen value and its associated eigen vector.

$$y = \frac{(Ax_{mean} + B) - (Ax)}{B} \tag{11}$$

3.2.3 RANSAC

RANSAC is one best curve fitting method used when the data contains outliers. This method enables us to reject necessary outliers while plotting the best fit.

RANSAC approach for estimation of coefficents:

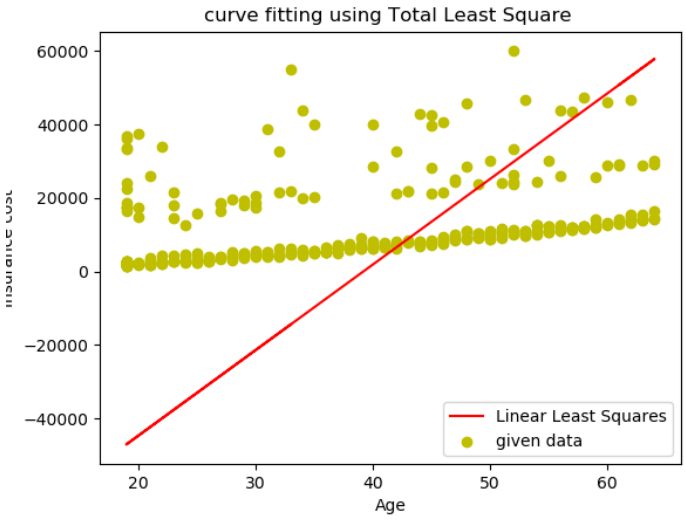

Figure 4: Curve Fitting using Total Least Square Method

Initially, from the given data set, we randomly pick a minimum of 2 points. These points will help us in defining a line.

Now, By drawing the line between randomly picked two points, we will try to calculate the perpendicular distance between them.

Next, we calculate the number of inliners by considering a minimum distance threshold, desired probability and two sdamples per iteration to cover large amount of data.

Depending on the threshold and number of inliners.This iteration continues untill best model is predicted.

3.2.4 Better choice of outlier rejection technique

1. In case of **least square**, The advantage is that this method works well when there arent ant outliers. However, when the data consists of outliers, this method is not suitable for best curve fitting as the curve shifts towards the outliers.
2. In case of **standard least sqaure**, the prime advantage is that the least square residue is smaller that the perpendicular distance between the points. However, When the data consists of outliers the curve is not the best fit.
3. In case of RANSAC, The major advantage is that the data works well with noise aswellas outliers.

Overall, I belive that RANSAC is one best technique among the three inorder best fit the data.

4 Problem 4

We know that SVD can be viewed as the generalized version of the Eigen-decomposition. Let A be a matrix of size m×n. The SVD of A is given by:

$$A = U \sum V^T \tag{12}$$

The matrix U (left singular values) of A gives us the eigenvectors of AAT. Similarly, as you expect, the matrix V (right singular values) of A gives us the eigenvectors of ATA. The non-zero singular values of A (found on diagonal entries of) are the square roots of non-zero eigenvalues of both AAT and ATA.

Procedure to solve SVD:

1. Calculate U matrix: By solving below equation and computing eigen vectors and normalization gives us U matrix

$$(A.A^T - \lambda * I) = 0 \tag{13}$$

2. Calculate V^T : By solving below equation and computing eigen vectors and normalization gives us V matrix

$$(A^T.A - \lambda * I) = 0 \tag{14}$$

3. Solve for Sigma by obtaining diagonal matrix.
Therefore, SDV is given by the equation:

$$A = U \sum V^T \tag{15}$$

Homography Matrix: Is obtained from last column of the V Matrix

$$\begin{bmatrix} 0.0531 & -0.049 & 0.061 \\ 0.077 & -0.093 & 0.0786 \\ 0.0236 & -0.0491 & 0.0762 \end{bmatrix}$$

5 Resource

5.1 My Github Repo

Code and Code instructions are provided under the readme section
go to the url: https://github.com/sumedhreddy90/ENPM673-Coursework

References

1. https://cmsc426.github.io/math-tutorial/#ransac
2. https://matplotlib.org/stable/gallery/lines_bars_and_markers/scatter_masked.html#sphx-glr-gallery-lines-bars-and-markers-scatter-masked-py
3. https://photo.stackexchange.com/questions/41273/how-to-calculate-the-fov-in-degrees-from-f
4. https://www.mathworks.com/discovery/ransac.html
5. https://en.wikipedia.org/wiki/Random_sample_consensus
6. https://vitalflux.com/ransac-regression-explained-with-python-examples/