ENPM673 HW01

PARAM DAVE
Master of Engineering in Robotics
University of Maryland, College Park
Email: pdave1@umd.edu

I. Problem 1

We have a camera with a resolution of 5MP where the camera sensor is square shaped with a width of 14mm and the focal length of the camera is 25mm.

A. Part 1

Field of View can be computed by the following equations

$$\phi = \tan^{-1}(\frac{d}{2f}) \tag{1}$$

$$AFOV = 2 * \phi \tag{2}$$

$$FOV = 2 * D * \tan(\frac{AFOV}{2}) \tag{3}$$

Here, sensor dimension d = 14 mm ; focal length f = 25 mm, using equation 1 we get ϕ = 31.284 °.
Using equation 2 we get FOV = 62.568°.
Using equation 3 and assuming distance of object from camera D = 20m we get Field of View = 24.3° in Horizontal and 24.3° in vertical direction.

Thus Field of view is 24.3° in both directions.

B. Part 2

To find minimum number of pixels that the object will occupy in the image, we first compute the distance at which the image will form. Using that distance we will find the height of the image and find the total area occupied by the square. Multiplying it by the pixel per mm we will get the pixel number. The equations for the said given below.

$$\text{Lens Formula}: \frac{1}{D} + \frac{1}{D'} = \frac{1}{f} \tag{4}$$

$$\text{Greens Equation}: \frac{y'}{y} = \frac{D'}{D} \tag{5}$$

Using equation 1 and value of f and D equal to 25mm and 20000mm we get D' = 25.03 mm. Using equation 2 and value of y = 50 mm we get y' =0.0625 mm.
Thus image area would be 0.0625 mm .0625 mm.
As camera has 5000000 pixels in 14 mm mm area , 1 mm × 1 mm will have $\frac{500000}{14*14}$ = 25510 pixels

Thus total pixels occupied by the image will be 25510 × 0.0625 × 0.0625 = 100 pixels.

II. Problem 2

Given a video of ball we try fit the best parabola using Standard Least Square Method. In the first step we would process the video frame by frame and process the frames to gray scale to find the contour which is the ball. Using contour, we find the center of the contour and save it in an array. We try to find the parabola parameters given in the equation 6 to best fit the curve

$$y = ax^2 + bx + c \tag{6}$$

Using the method of Least Square we compute the parameter matrix as given in equation 7.

$$B = (X^TX)^{-1}(X^TY) \tag{7}$$

Plotting the values of x and y calculated using the parameter matrix we get the parabolas as shown in figures 1 and figure 2. The first figure shows parabola fitted to ball path with no noise and the second figure shows parabola fitted to ball path with noise.

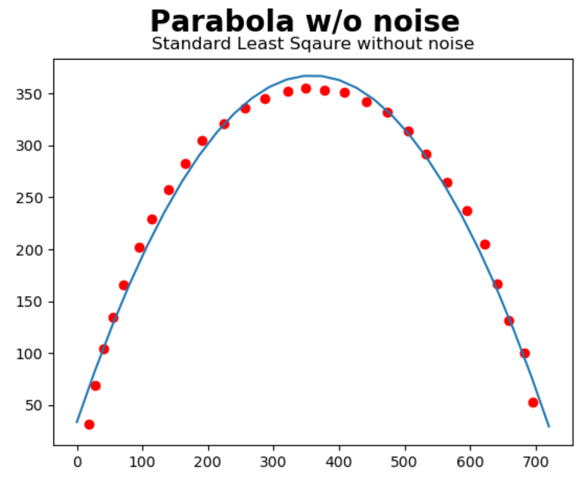

Fig. 1. Parabola curve fitting using Standard Least Square without any noise

III. Problem 3

In this problem we try to fit a data of health insurance costs based on the person's age. We will use Standard Least Square , Total Least Square and RANSAC algorithms to fit the given data and try comparing the three methods.

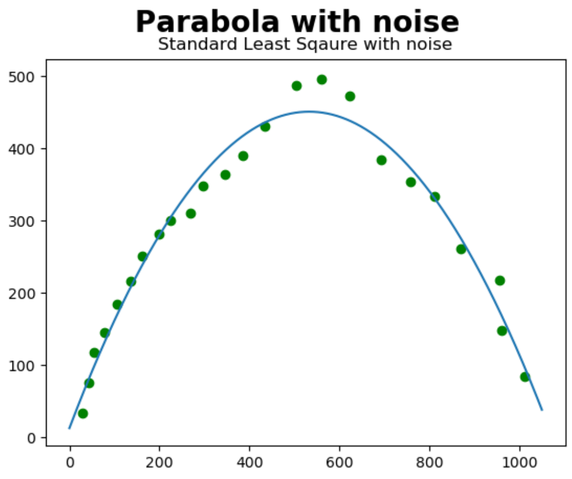

Fig. 2. Parabola curve fitting using Standard Least Square with noise

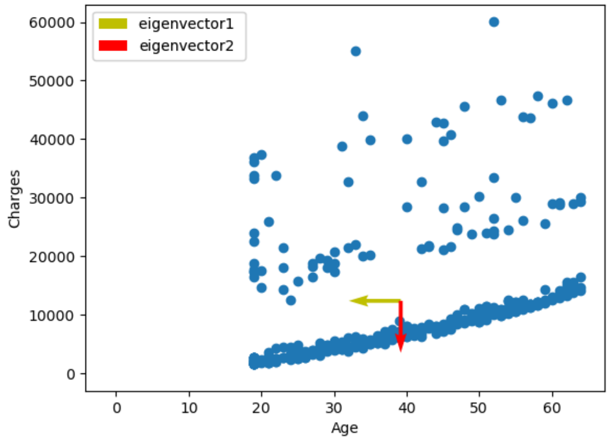

Fig. 3. Age vs Charge

A. Part 1

The data given is the age of the patient and the cost of insurance. Covariance matrix without normalizing the age and cost can be computed as following equation 8 and equation 9

$$Cov_{xy} = \Sigma((X - \overline{X})(Y - \overline{Y}))/N \tag{8}$$

$$\text{Covariance Matrix} = \begin{bmatrix} Cov_{xy} & Cov_{yy} \\ Cov_{xx} & Cov_{yx} \end{bmatrix} \tag{9}$$

Following matrix shows Covariance matrix between age and charge.

$$\begin{bmatrix} 195.84 & 52370.7 \\ 52370.7 & 122218099 \end{bmatrix} \tag{10}$$

Calculated the eigenvectors of the Covariance matrix and plotted them on Age vs Charges in figure 3.

For normalized age and cost of insurance we can compute the covariance matrix from the equation 8. and 9. The eigenvectors of the normalized covariance matrix are plotted on Age vs Charges plot in figure 4 below.

The normalized covariance matrix is shown in equation 11.

$$\text{Normalized Covariance Matrix} = \begin{bmatrix} 0.0975 & 0.01998 \\ 0.01998 & 0.03594 \end{bmatrix} \tag{11}$$

B. Part 2

We are given a data of age vs insurance cost and are required to use Standard Least Square , Total Least Square and RANSAC algorithms to find the best line that represents the data. The solution to Standard Least Square Curve fitting is shown in figure 5 and that of Total Least Square and RANSAC are shown in figure 6 and figure 7 respectively.

Advantages of Standard Least Square.
1) Easy to understand and implement.

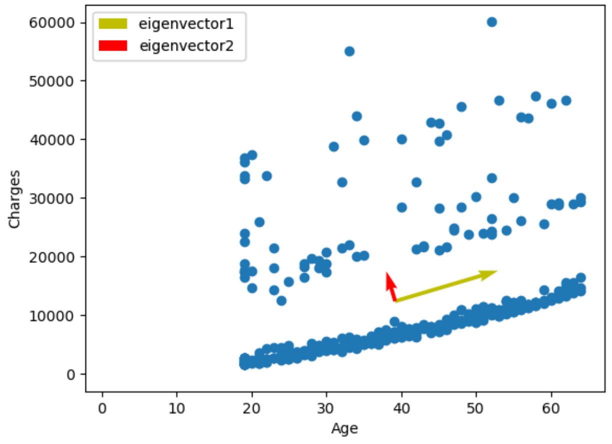

Fig. 4. Age vs Charge for noralized covariance

2) Has a wide range of applications.
3) Computationally cheap
4) No starting values are needed.

Disadvantages of Standard Least Square.
1) Sensitivity to outliers. One or two extreme outliers could skew the parameters.
2) Prone to over-fitting.
3) Poor extrapolation properties
4) Has Attenuation Bias

Advantages of Total Least Square:
1) Reduces Attenuation bias.
2) Accounts for noise in both dependent and independent variables.
3) No starting values are needed.

Disadvantages of Total Least Square:

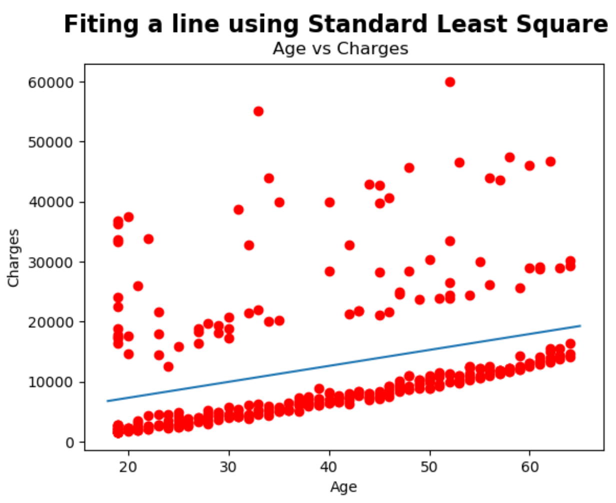

Fig. 5. Standard Least Square line fitting to Age vs Charge

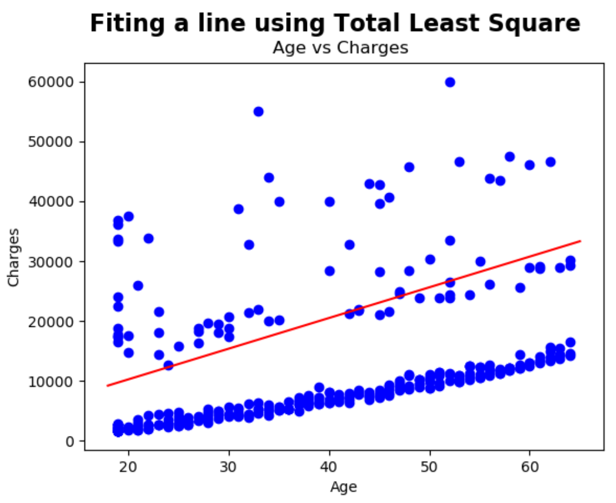

Fig. 6. Total Least Square line fitting to Age vs Charge

1)TLS estimation depend on the units in which variables are measured.
2)Sensitive to outliers.

Advantages of RANSAC:
1) It can estimate the parameters with a high degree of accuracy even when a significant number of outliers are present in the data set.
2) A general method that can be applied most of the cases
3) Fast in huge data sets

Disadvantages of RANSAC:
1)Computationally Expensive.
2)Too many parameters to tune.
3)Correct values of parameters are required by trial and error method.
4) Requires prior knowledge about data regarding model to fit.

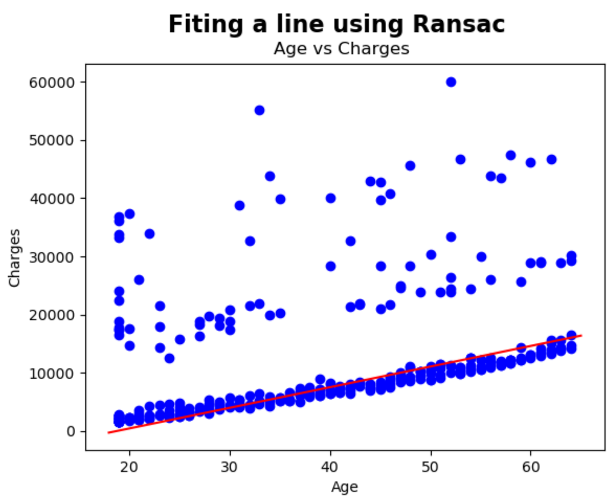

Fig. 7. RANSAC fitting to Age vs Charge

C. Part 3

1) Standard Least Square Method: Standard Least Square calculated the squared distance between the variable and predicted dependent variable and tries to minimizes it. In the solution let X and Y be independent and dependent set of variables. We try to find the solution of the equation 12 where B is the vector of parameters of the linear function we are trying to predict. Example for a line whose equation is defined by equation 13, B contains values of m and c.

The solution for B by given in equation 14.

$$XB = Y \tag{12}$$

$$y = mx + c \tag{13}$$

$$B = (X^T X)^{-1} X^T Y \tag{14}$$

The implementation of Standard Least Square is given in least_square.py which contains two function each calculates Standard Least Square for parabola and line respectively using equations 12-14.

2) Total Least Square Method: Total Least Square Method is a least squares data modeling technique in which observational errors on both dependent and independent variables are taken into account. It measures error perpendicular to the line or curve needed to fit. Assume X and Y are independent and dependent variable,we set Z as [X Y]. Calculating singular value decomposition (will discuss in next section) of the Z matix we extract the V matrix and slice is where V_{xy} and V_{yx} are given by equation 15 and 16 respectively

$$V_{xy} = V[: n, n :]; \text{n is the number of elements in X matrix} \tag{15}$$

$$V_{yy} = V[n :, n :]; \text{n is the number of elements in X matrix} \tag{16}$$

As in Standard Least Square method we computer B , here as well we compute B given by equation 16. We use B to calculated the line/curve. The implementation of Total Least Square is given in tls.py file.

$$B = -V_{xy}/V_{yy} \tag{17}$$

3) RANSAC: RANSAC stands for Random sample consensus is an iterative voting algorithm for estimating a mathematical model from a data set that contains outliers. The RANSAC algorithm works by identifying the outliers in a data set and estimating the desired model using data that does not contain outliers.

The input to the RANSAC algorithm is a set of observed data values, a way of fitting some kind of model to the observations, and some confidence parameters. RANSAC achieves its goal by repeating the following steps:

1)Select a random subset of the original data. Call this subset the hypothetical inliers.
2)A model is fitted to the set of hypothetical inliers.
3)All other data are then tested against the fitted model. Those points that fit the estimated model well, according to some model-specific loss function, are considered as part of the consensus set.
4)The estimated model is reasonably good if sufficiently many points have been classified as part of the consensus set.
5)Afterwards, the model may be improved by reestimating it using all members of the consensus set.

This procedure is repeated a fixed number of times, each time producing either a model which is rejected because too few points are part of the consensus set, or a refined model together with a corresponding consensus set size. In the latter case, we keep the refined model if its consensus set is larger than the previously saved model.

The number of iterations k, can be determined as a function of the desired probability of success p using a theoretical result and is shown in equation 18 Let p be the desired probability that the RANSAC algorithm provides at least one useful result after running. RANSAC returns a successful result if in some iteration it selects only inliers from the input data set when it chooses the n points from which the model parameters are estimated. Let w be the probability of choosing an inlier each time a single point is selected given by equation 19.
Assuming that the n points needed for estimating a model are selected independently, w^n is the probability that all n points are inliers and $1 - w^n$ is the probability that at least one of the n points is an outlier, a case which implies that a bad model will be estimated from this point set. That probability to the power of k is the probability that the algorithm never selects a set of n points which all are inliers and this must be the same as $1 - p$.

$$k = \frac{log(1-p)}{log(1-w^n)} \tag{18}$$

$$w = \text{number of inliers in data / number of points in data} \tag{19}$$

Implementation of RANSAC is given in ransac.py.All parameters are pre calculated and only inputs are the X and Y variables. Standard Least Square method was used to fit a line.

While performing the curve fitting it was observed that Total Least Square was the most prone to outliers followed by Standard Least Square and RANSAC. RANSAC performed the best out of the three when there were outliers and was also difficult to execute.

IV. Problem 4

In linear algebra, the Singular Value Decomposition (SVD) of a mxn matrix A is a factorization of that matrix into three matrices given by equation 19. Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and A^TA. The eigenvectors of A^TA make up the columns of V , the eigenvectors of AA^T make up the columns of U. Also, the singular values in S are square roots of eigenvalues from AA^T or A^TA. The singular values are the diagonal entries of the S matrix and are arranged in descending order.

$$A = U\Sigma V^T \tag{20}$$

Here U is mxn matrix of the orthonormal eigenvectors of AA^T V^T is transpose of a nxn matrix containing the orthonormal eigenvectors of A^TA. Σ is a nxn diagonal matrix of the singular values which are the square roots of the eigenvalues of A^TA.

Eigenvalues λ are calculated using equation 21.

$$det(A - \lambda I) = 0 \tag{21}$$

Similarly eigenvalues v_i corresponding to each eigenvalue λ_i is given by equation 22.

$$det(A - \lambda_i).v_i = 0 \tag{22}$$

Finally Homography matrix is the last row of V^T computer using SVD described above. The vector is reshaped into a 3x3 matrix which is shown in the equation 23.

$$H = \begin{bmatrix} 5.31056351e^{-2} & -4.91718843e^{-3} & 6.14648552e^{-1} \\ 1.77018784e^{-2} & -3.93375075e^{-3} & 7.86750146e^{-1} \\ 2.36025045e^{-4} & -4.91718843e^{-5} & 7.62164205e^{-3} \end{bmatrix} \tag{23}$$

References

[1] "Total least squares - people.duke.edu." [Online]. Available: https://people.duke.edu/ hp-gavin/SystemID/CourseNotes/TotalLeastSquares.pdf.

[2] "1 singular values - university of California, Berkeley," SVD Notess. [Online]. Available: https://math.berkeley.edu/ hutching/teach/54-2017/svd-notes.pdf. .

[3] "Overview of the RANSAC algorithm - York University," Overview of the RANSAC algorithm . [Online]. Available: http://www.cse.yorku.ca/ kosta/CompVis_Notes/ransac.pdf.

[4] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. Cambridge, Mass: The MIT Press, 2017.

[5] R. Hartley and A. Zisserman, Multiple view geometry in Computer Vision. Cambridge: Cambridge University Press, 2019.