Name: Karam Mawas 1.Feb.2016

Matriculation number: 2946939

Topology and Optimization

Exercise 2

Determination of a Regression Line using Total Least Squares

Generate points on a specific line with noise. The Task is to fit these noisy points by line regression using two different methods.

- **Least square:**
  The Least square model is a method used to adjust the observations in one direction. Either on x, y- direction. Which means the erroneous vector will follow the observation direction. On other words, will not be optimal. Due to the fact, the erroneous vector should be the smallest value which it would be in geometry as the perpendicular distance to the target.
  The model:

$$l = A.x + \varepsilon, A_{n\times 2} = \begin{bmatrix} x_i & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}; n = 21$$

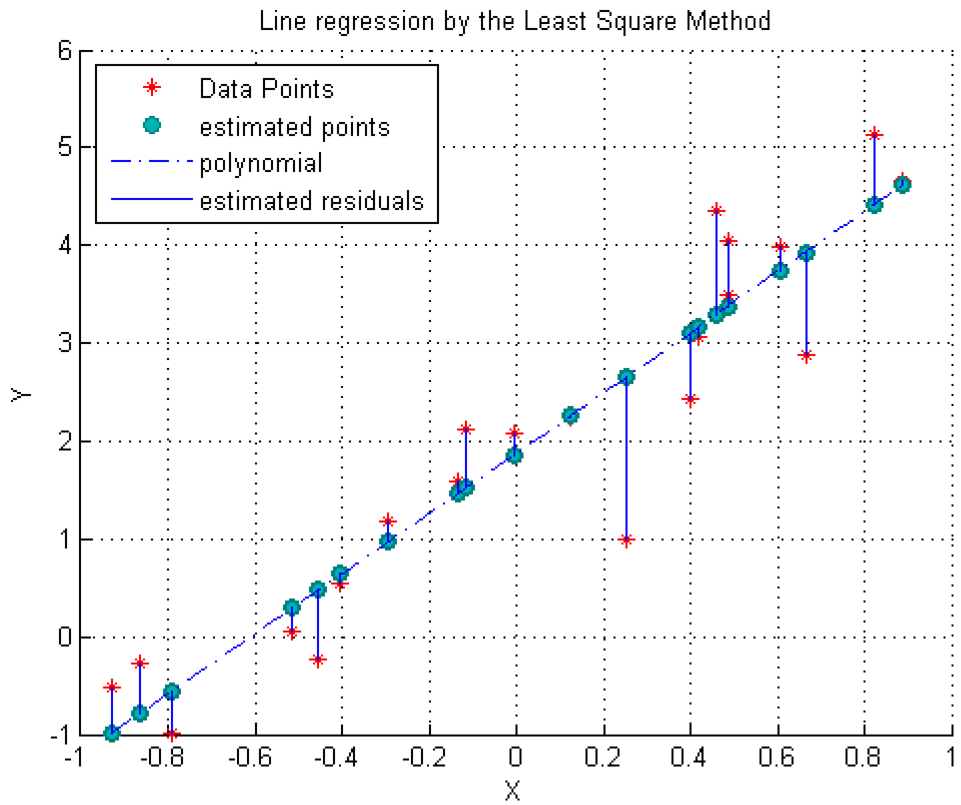

Fig-1: Line regression by the Least Square Method.

The figure shows the erroneous vectors follow the observations axis which in this case is the y-axis.

Matriculation number: 2946939

- **Total Least Square:**

The methodology depends on the Singular Value Decomposition. Initializing the problem depends on the same assumption of the Least Square $A.\xi - l \cong 0$; $\xi = \begin{bmatrix} a \\ b \end{bmatrix}$

Rearrange the previous equation in order to be written on a matrices multiplication.

$$[A \quad l].\begin{bmatrix} \xi \\ -1 \end{bmatrix} \approx 0$$

Here in this equ. The solution space has been augmented by the observation vector. On other words, the A matrix considered to contain errors. The ***'-1'*** is the condition, which will bring the observations to be adjusted. The SVD has been applied on the augmented solution space. The sub-matrices U and V are the eigen-vectors of the main one and the S matrix is the eigen-values. On the matrix S, the element $S_{3,3}$set to be zero. The adjusted matrix of the augmented solution space can be computed as follow:

$$[\hat{A} \quad \hat{l}] = U\hat{S}V^T$$

The adjusted observation vector on y- direction can be computed from the following equ.

$$\hat{l} = \hat{A}.\hat{\xi}$$

But we have to consider our first constraint. The matrix V is a three by three matrix. The last column contains the unknown elements. By using the Matlab code *'reshape(V(:,3),3,1)'.* The vector of interests has been extracted now the constraints can be applied by dividing the negative value of the last element of the extracted vector over the vector itself.

$$\begin{bmatrix} \underline{\xi} \\ -1 \end{bmatrix} = \frac{-1}{V_{3,3}}.\underline{V}$$

Now the observations on the x-direction should also be computed. The line equation is as the following:

$$y_i = a.x_i + b \text{ , } A = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \hat{\xi} = \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix}$$

So the observation on x should be the first column of the matrix A. But if we take a look on the second column, it has to be a unite vector. The estimated 'A' matrix does not contain ones on the second column, due to a certain scale. The observations on x can be computed from the estimated observations on y-direction.

$$\underline{\hat{x}} = \frac{\underline{\hat{y}} - \hat{a}}{\hat{b}}$$

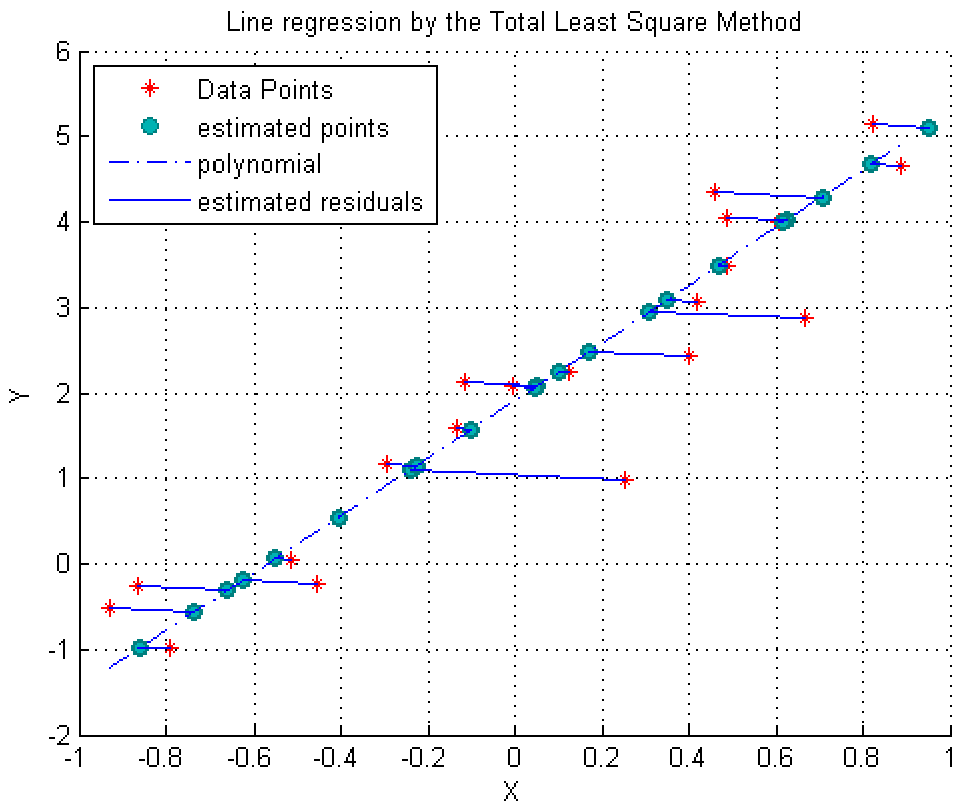

Fig-2: Line regression by the Total Least Square Method

The erroneous vectors do not follow either the x or the y direction. Which is definitely better than the previous method. The figure shows that the erroneous vectors are not orthogonal on the regression line due to the fact, that this points are not the true values they are estimated ones. Also I think the reason behind the non-orthogonality or this big difference. That we did not consider the errors on the unknowns or the non-linear component on the line equation. $y_i = a.x_i + b$

The term $a.x_i$ is not linear if we consider the erroneous on x and a. which I think is worth to compare between the Total Least Square Method and the Gauß-Helmert model.

- **Gauß-Helmert model:**
  This model is also called the mixed model because it combine the A-model and the B-model. In our example we putted obstacles on both x and y observations.

A-Model:

$$y_i = a.x_i + b + e_{y_i}$$

B-Model:

$$\frac{y_1 - e_{y_1} - (y_2 - e_{y_2})}{y_1 - e_{y_1} - (y_3 - e_{y_3})} = \frac{x_1 - x_2}{x_1 - x_3}$$

Mixed model

$$y_i = a.(x_i - e_{x_i}) + b + e_{y_i}$$

The unknowns here are 'a, b, e_{x_i}, e_{y_i}'

The linearization approach should be proceed.

Taylor points: $a = a_{TP} + \Delta a\,, b = b_{TP} + \Delta b\,, e_{x_i} = e_{x_i}^{TP} + \Delta e_{x_i}\,,\ e_{y_i} = e_{y_i}^{TP} + \Delta e_{y_i}$

The $e_{x_i}^{TP}$ and $e_{y_i}^{TP} = 0$ on the first iteration. The a_{TP} and b_{TP} are computed as follows:

I computed them from the first two given points. Because they should fulfill the line equation. By this I should be in the vicinity of the right values.

$$a_{TP} = \frac{y_1 - y_2}{x_1 - x_2}\ , b_{TP} = \frac{y_2 . x_1 - y_1}{x_1 - x_2}$$

$$f\left(\mathrm{a}, \mathrm{b}, e_{x_i}, e_{y_i}\right) = f\left(\mathrm{a}_{TP}, \mathrm{b}_{TP}, e_{x_i}^{TP}, e_{y_i}^{TP}\right) + \left.\frac{\partial f}{\partial a}\right|_{TP} (a - \mathrm{a}_{TP}) + \left.\frac{\partial f}{\partial b}\right|_{TP} (b - \mathrm{b}_{TP})$$
$$+ \left.\frac{\partial f}{\partial e_{x_i}}\right|_{TP} \left(e_{x_i} - e_{x_i}^{TP}\right) + \left.\frac{\partial f}{\partial e_{y_i}}\right|_{TP} \left(e_{y_i} - e_{y_i}^{TP}\right) + \cdots (H.O.T)$$

$$\Rightarrow \underbrace{f\left(\mathrm{a}_{TP}, \mathrm{b}_{TP}, e_{x_i}^{TP}, e_{y_i}^{TP}\right) - \left.\frac{\partial f}{\partial e_{x_i}}\right|_{TP} \left(e_{x_i}^{TP}\right) - \left.\frac{\partial f}{\partial e_{y_i}}\right|_{TP} \left(e_{y_i}^{TP}\right)}_{W_i} + \underbrace{\left.\frac{\partial f}{\partial a}\right|_{TP} (\Delta a) + \left.\frac{\partial f}{\partial b}\right|_{TP} (\Delta b)}_{A_i}$$
$$+ \underbrace{\left.\frac{\partial f}{\partial e_{x_i}}\right|_{TP} \left(e_{x_i}\right) + \left.\frac{\partial f}{\partial e_{y_i}}\right|_{TP} \left(e_{y_i}\right)}_{B_i^T}$$

$$\Rightarrow W_i + A_i \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix} + B_i^T \begin{pmatrix} e_{x_i} \\ e_{y_i} \end{pmatrix} = 0\ ; W_i = y_i - (a_{TP} + b_{TP} . x_i)$$

$$A_i = -\begin{bmatrix} 1 & x_i - e_{x_i}^{TP} \end{bmatrix}, B_i^T = \begin{bmatrix} b_{TP} & -1 \end{bmatrix}$$

$$\begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix}_{n,1} + \begin{bmatrix} -1 & -(x_1 - e_{x_1}^{TP}) \\ \vdots & \vdots \\ -1 & -(x_n - e_{x_n}^{TP}) \end{bmatrix}_{n,2} \begin{pmatrix} \Delta a \\ \Delta b \end{pmatrix}_{2,1} + \begin{bmatrix} b_{TP} & \cdots & 0 & -1 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & b_{TP} & 0 & \ldots & -1 \end{bmatrix}_{n,2n} \begin{pmatrix} e_{x_i} \\ e_{y_i} \end{pmatrix}_{2.1} = [0]_{n,1}$$

The model now is:

$$W_i + A_i . \Delta\xi + B_i^T . e = 0$$

To compute the estimated values we should the target function should be minimum.

$$\mathcal{L}(e, \Delta\xi, \lambda) = \frac{1}{2} e^T I e + \lambda^T (W + A . \Delta\xi + B^T . e) \longrightarrow \min(e, \Delta\xi, \lambda)$$

$$\begin{bmatrix} I_{2n,2n} & 0_{2n,t} & B_{2n,n} \\ 0_{t,2n} & 0_{t,t} & A_{t,n}^T \\ B_{n,2n}^T & A_{n,t} & 0_{n,n} \end{bmatrix}_{3n+t,3n+t} \begin{bmatrix} \widehat{e_{2n,1}} \\ \widehat{\Delta\xi_{t,1}} \\ \widehat{\lambda_{n,1}} \end{bmatrix}_{3n+t,1} = \begin{bmatrix} 0_{2n,1} \\ 0_{t,1} \\ -W_{n,1} \end{bmatrix}_{3n+t,1} ; n = 21, t = 2$$

$$\begin{bmatrix} \widehat{e_{2n,1}} \\ \widehat{\Delta\xi_{t,1}} \\ \widehat{\lambda_{n,1}} \end{bmatrix}_{3n+t,1} = \begin{bmatrix} I_{2n,2n} & 0_{2n,t} & B_{2n,n} \\ 0_{t,2n} & 0_{t,t} & A_{t,n}^T \\ B_{n,2n}^T & A_{n,t} & 0_{n,n} \end{bmatrix}^{-1} . \begin{bmatrix} 0_{2n,1} \\ 0_{t,1} \\ -W_{n,1} \end{bmatrix}$$

The thresholds are: $\|\hat{e} - e_{TP}\| < \varepsilon, \|\widehat{\Delta\xi}\| < \varepsilon$ (iteration until this constraints will be fulfilled!).

I applied this on Matlab. Unfortunately, I made mistake on the while loop (I think!). But I believe this model will be the most accurate among the two others. And the erroneous vectors would be almost perpendicular to the line regression. But this model has dis-advantage comparing with the Total Least Square. As it shown here the matrices expand dramatically comparing with the initial ones. The Total Least square is faster due to the computation time. Another point, in the Mixed-model, it is not robust as the Total Least Square. Referring to the estimation equ. We have to inverse the matrix which sometimes, we could face ranking problem.