Linear regression using robust functions

Lluís Garrido – lluis.garrido@ub.edu

October 2021

Abstract

This laboratory is focused on the principles of data regression. We will focus on some of the well known methods and on the importance of the error function used to perform the regression to ensure robustness to outliers. Notice that this lab is still focused on unconstrained optimization.

1 Introduction

Linear regression is an approach to model the relationship between two continuous, quantitative, variables. One variable, denoted x, is regarded as the predictor or independent variable. The other variable, denoted as y, is regarded as the response or dependent variable.

Take a look at figure 1. We may see a set of data points in blue. In this example the objective is to approximate the data points using a simple linear regression, i.e. we only have one predictor variable.

What is the best fitting line? Let $\mathbf{x}_i = \{x_i, y_i\}$ with $i = 1 \dots m$ be the data points. There are different approaches that one may take to compute the best fitting line, see figure 2. The figure shows two different methods: the standard fitting using vertical offsets, and the fitting using perpendicular offsets.

We focus here with the standard approach since it is easier to formulate and implement. Let $\hat{y}_i$ be the fitted value for element i. The equation for the best fitting line is $\hat{y}_i = w_0 x_i + w_1$, where $w_0 \in R$ and $w_1 \in R$ are the parameters to be estimated. The parameters w_0 and w_1 may be computed in such a way that the prediction error (or residual error), $e_i = |\hat{y}_i - y_i|$, is minimized. A line that fits the data "best" will be one for which the m prediction errors are as small as possible in some overall sense. Observe that in this case we perform a fitting using vertical offsets, see figure 2. There are other kind of approaches that use the perpendicular offset which is not the focus in this lab.

A classical way to proceed to compute the optimal parameters w_0 and w_1 is to use the least squares error function, that is,

$$Q = \frac{1}{2}\sum_{i=1}^{m} e_i^2 = \frac{1}{2}\sum_{i=1}^{m} (\hat{y}_i - y_i)^2 \tag{1}$$

One may use the gradient (or Newton) descent method, for instance, to obtain the optimal values for w_0 and w_1. Let us see how to proceed in this case

$$Q = \frac{1}{2}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{m} (w_0 x_i + w_1 - y_i)^2$$

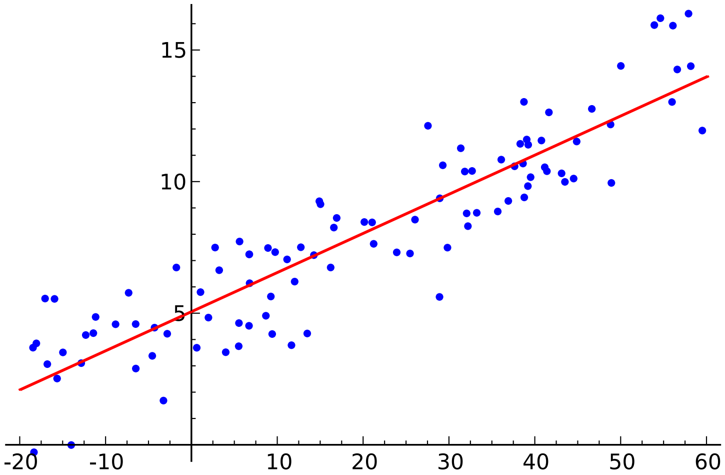

Figure 1: Simple linear regression example. Data points are marked in blue and the objective is to model the points with a linear model. Obtained from https://en.wikipedia.org/wiki/Linear_regression.

We want to obtain the values of w_0 and w_1 that minimize Q. For this purpose let us compute the corresponding gradient

$$\begin{aligned}\frac{\partial Q}{\partial w_0} &= \sum_{i=1}^{m} (w_0 x_i + w_1 - y_i)\, x_i \\ \frac{\partial Q}{\partial w_1} &= \sum_{i=1}^{m} (w_0 x_i + w_1 - y_i)\end{aligned}$$

The parameters to be estimated are $\mathbf{w} = (w_0, w_1)$. The gradient descent is

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \alpha^k \nabla Q(\mathbf{w}) \qquad \nabla Q(\mathbf{w}) = \left(\frac{\partial Q}{\partial w_0}, \frac{\partial Q}{\partial w_1}\right)$$

Experiments to perform

You are proposed to follow the next steps:

1. Implement the proposed method using gradient descent with backtracking (or a small constant α value). You may check your method with a randomly generated set of points

```
m = [0.,0.]
angle = 45*math.pi/180
rot = np.array([[math.cos(angle), -math.sin(angle)], [math.sin(angle),
math.cos(angle)]])
lamb = np.array([[100,0],[0,1]])
s = np.matmul(rot, np.matmul(lamb, rot.transpose()))
c = np.random.multivariate_normal(m,s,100)
```

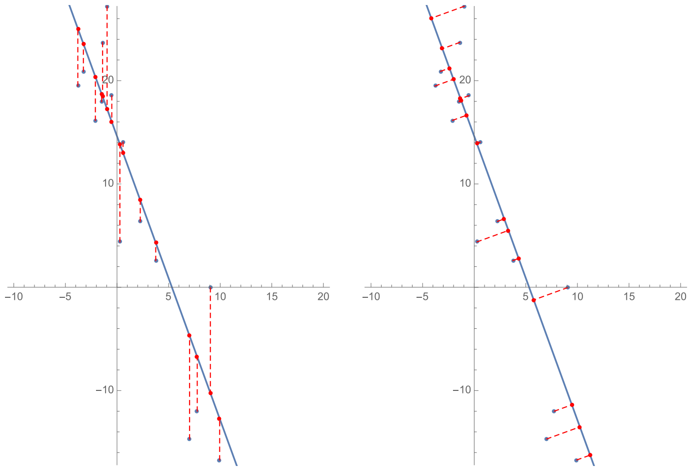

Figure 2: Left: fitting using vertical offsets, Right: fitting using perpendicular offsets. Obtained from http://toddreed.name/articles/line-fitting/.

The `angle` value as well as the `lamb` matrix allows us to control the shape of the random values that are generated.

Compute the parameters associated to the model and draw the lines that is obtained with the set of points.

2. Let us now check the sensitivity of the method to outliers, that is, points that do not follow the model. Using a set of points generate with an angle of 45 degrees, change the value of one point to a value "far away" from the set of points, for instance `c[1] = [-40,20]`. Draw the line that approximates the set of points and observe that one point may have a large influence in the obtained solution.

   Try to change other points of the original data set to see which is the influence of outliers. You may, for instance, randomly change some points of the original data set with may the far or close away from the original distribution of the dataset.

Outliers may be usual when performing data analysis. Therefore, it is important to use robust functions to outliers in order to obtain the expected result.

2 Robust functions

The least squares method is known to be sensitive to outliers, i.e. samples that do not fit the model (since they may be noisy samples, for instance). The reason is due to the fact that the least squares method minimizes the squared error which may be very large for an outlier. The outliers may thus have a large influence in the numerical values of the parameters to be estimated.

We need to modify equation (1) so as to make it more robust to outliers. Without entering into exhaustive details, a way to proceed is to minimize

$$\sum_{i=1}^{m} \rho(e_i)$$

where $\rho(u)$ is a robust error function. For the least squared method $\rho(u) = \frac{1}{2}u^2$, but one may take other functions such as the Cauchy function, which is defined to be[1]

$$\rho(u) = \frac{c^2}{2} \log \left[1 + \left(\frac{u}{c} \right)^2 \right] \tag{2}$$

where $c \in R$.

Experiments to perform

Assume, for simplicity, that $c = 1$ is taken for the Cauchy function (you may use the set of points with a 45 degrees)

1. Plot the least squares function, $\rho(u) = \frac{1}{2}u^2$, and compare it with the Cauchy function, Eq. (2), in order to see the "importance" that is given to each prediction error u. You may, for instance, plot the function $\rho(u)$ for $|u| \leq 10$.
2. Implement the algorithm that allows to compute the parameters w_0 and w_1 using the Cauchy function. For that issue you can use the backtracking gradient descent method (there is no need to use the Newton method).
3. Compare the results you obtain with the least squares function and the Cauchy function assuming that no outliers are in the dataset.
4. Compare now the results with only one outlier. You may proceed as has been proposed before. The Cauchy function should be more robust than the quadratic function.
5. Test the influence of the parameter c in the obtained regression. You may, for instance, check the results obtained with $c = 1$, $c = 100$, $c = 1/100$ and $c = 1/1000$. Can you reason why these results are obtained? To do this, you are recommended to compare the corresponding $\rho(u)$ functions.
6. The Cauchy function is not "perfect", it is not robust for any number of outliers. Using $c = 1$ and $c = 1/100$ you may, as before, gradually introduce more number of outliers into the dataset. You should see that for a certain number of outliers, the the Cauchy function will be sensitive to the "high" number of outliers. Can you comment on the experiments you have performed?

Report

You are asked to deliver a report (PDF, notebook, or whatever else you prefer) that can be done in *pairs of two persons* (or *individually*). Comment each of the steps you have followed as well as the results and plots you obtain. Do not expect the reader (i.e. me) to interpret the results for you. I would like to see if you are able to understand the results you have obtained.

If you want to include some parts of code, please include it within the report. Do not include it as separate files. You may just deliver the Python notebook if you want.

[1] If you are interested in more information on this issue you may begin with the reference Steward, C. "Robust parameter estimation in computer vision", SIAM Review, Vol. 41, No. 3, pp. 513–537, 1999.