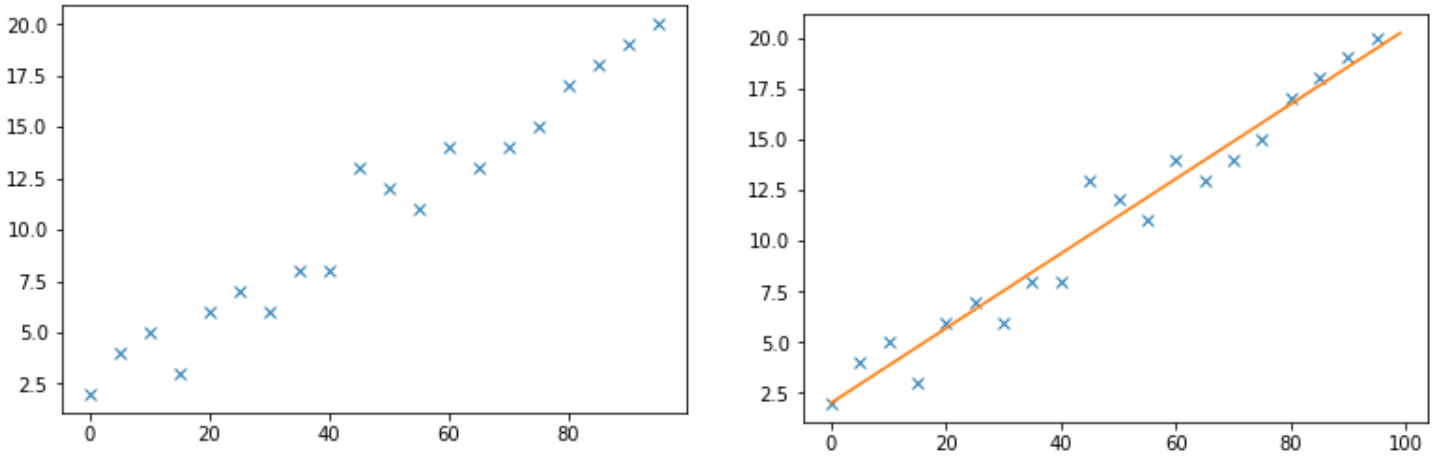
Linear Regression

Khaled GRIRA

Motivation of the study

We want a straight line that would fit our data the best way possible so as to make predictions on values that are not in the training set.



Exemple in two dimensions,that's a pretty good approximation

Equation setting

Our hypothesis is linear. $h(x) = w^T x = x^T w$ with $(w, x) \in \mathbb{R}^n$ X= $\begin{pmatrix} x_1^T \\ . \\ . \\ x_M^T \end{pmatrix}$ and Y= $\begin{pmatrix} y_1 \\ . \\ . \\ y_M \end{pmatrix}$ our training set. We want our straight line to fit the data as well as possible so we want the averange distance between every point and the straight line to be as little as possible. To make it easier we'll take the average of the squared distance (Mean square error)

Our goal is to minimize $\mathcal{L}(w) = \frac{\|h(X) - Y\|^2}{M}$ with respect to w h(X) := $\begin{pmatrix} h(x_1) \\ . \\ . \\ h(x_M) \end{pmatrix}$. Where the norm is the Euclidean norm, which will be differentiable.

Solving linear regression

Two ways to prove the normal equation, our function is convex and positive we just need to find where the gradient is equal to zero or we can use the fact that the norm is the smallest when Xw is equal to the projection of Y on Vect(X).

We can see that $Xw = \begin{pmatrix} x_1^T w \\ . \\ . \\ x_M^T w \end{pmatrix} = h(X)$. Then $\mathcal{L}(w) = \frac{\|Xw - Y\|^2}{M}$, $\mathcal{L}$ is differentiable.

$\|Xw - Y\|^2 = \|Xw\|^2 - 2 < Xw, Y > + \|Y\|^2$ so $\frac{\partial \mathcal{L}}{\partial w_{i_0}}(w) = \frac{2}{M} \sum_{k=1}^{M} X_{k,i_0}(w^T x_k - Y_k)$.

In the end we have $\frac{\partial \mathcal{L}}{\partial w} = X^T(Xw - Y)$, for a local optimum $\frac{\partial \mathcal{L}}{\partial w} = 0$ and given that our function is convex and positive, a local optimum is necessarily a global minimum. w, where $\mathcal{L}$ is at its minimum, verifies $X^T X w = X^T Y$

$\mathcal{L}(w)$ is minimal for $w = (X^T X)^{-1} X^T Y$. If $X^T X$ is not invertible, you can find the minimum with gradient descent or by using the generalized inverse. Gradient descent, or some other optimization algorithm, should be chosen over the normal equation whenever we have a large dataset or a lot of features.

The function lstsq from numpy.linalg in Python automatically computes the solution. If it is in two dimensions, just use linregress from scipy.stats which will give you the slope and intercept.