Least squares regression, overdetermined linear systems and Moore-Penrose pseudo-inverse

Kamila Zdybał

Université libre de Bruxelles, kamila.zdybal@ulb.ac.be
camillejr.github.io/science-docs, kamila.zdybal@gmail.com

Preface

The goal of this paper is to explain why Moore-Penrose pseudo-inverse give a least squares regression. And as we travel the journey to get there, I would like to give a handful of insights along the way.
This document is still in preparation. Please feel free to contact me with any suggestions, corrections or comments.

Keywords

overdetermined linear systems, least squares regression, Moore-Penrose pseudo-inverse, linear algebra

Contents

1 Overdetermined linear systems

When we want to fit an n-dimensional straight line through the data, n number of points is sufficient to construct that line.
For instance, for a two dimensional case we have a system of equations:

$$\begin{cases} y_1 = ax_1 + b \\ y_2 = ax_2 + b \end{cases} \tag{1}$$

that we can also write in a matrix form:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \cdot \begin{bmatrix} b \\ a \end{bmatrix} \tag{2}$$

or in short:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A} \tag{3}$$

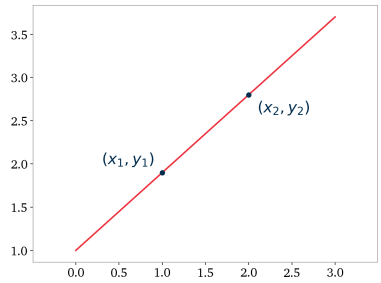

Figure 1: Solution to a 2-dimensional linear system.

Whenever $N > n$ points are provided, we are dealing with an overdetermined linear system. We have redundant data that does not bring any new information to the system. Once we have solved this system, its parameters a and b do not get updated with any new data point. Such system can still be described by the general eq.(3), except now we include all N points of our data:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{bmatrix} \cdot \begin{bmatrix} b \\ a \end{bmatrix} \tag{4}$$

Notice here that the matrix $\boldsymbol{X}$ has n columns and the first column is padded with ones to account for the constant term b. We could also write $\boldsymbol{Y} = \boldsymbol{x}\boldsymbol{A} + \boldsymbol{b}$ if we wanted to separate the constant term. That is something to keep in mind when you write your regression code.

Can we go further? What about the data that is in excess ($N > n$) and additionally, it is not co-linear - a straight line cannot be fitted to pass through all of these data points? In that case, the overdetermined linear system does not have a unique solution. A solution $\boldsymbol{A}$ to a problem $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}$ does not exist. In particular, different a and b might be needed to transform point x_1 to y_1 and x_2 to y_2 then to transform point x_8 to y_8 and x_9 to y_9.

But even for such system we are naturally dragged to taking a pen and drawing a straight line that passes *more or less* through that cloud of points. That fitting is a **linear regression**. How we deal with the *more or less* is a matter of what regression technique we choose.

Drawing the line that follows the directionality of that cloud of points seems like an easy task to do by hand. But how do we automate this process? How do we find that line mathematically, if we cannot solve $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}$ directly? That's where the fun starts.

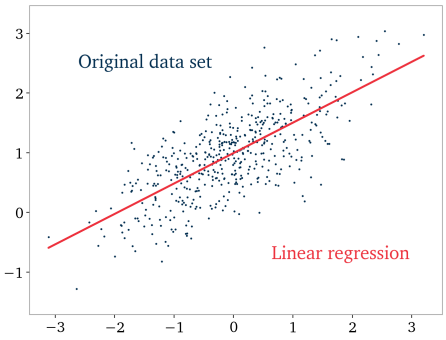

Figure 2: Example of a linear regression.

2 Least squares

One way to perform linear regression is to use least squares (LS). (And really I would like to stress that it is just one way to do it, there can be many more.) In other words, least squares regression is a certain way to estimate a and b or to refer to the general case, it is a certain way to estimate the matrix $\boldsymbol{A}$.

2.1 Derivation

$$\boldsymbol{X}^T(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{A}) = 0 \tag{5}$$

$$\boldsymbol{A} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{6}$$

Predictions for the existing values of $\boldsymbol{X}$ are then:

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\boldsymbol{A} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \tag{7}$$

Why A contains the coefficient for the *best-fit* linear function?

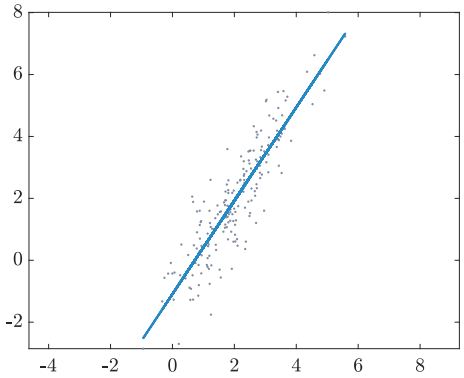

Figure 3: Linear basis function LS regression.

2.2 Predictions

2.3 Basis function methods

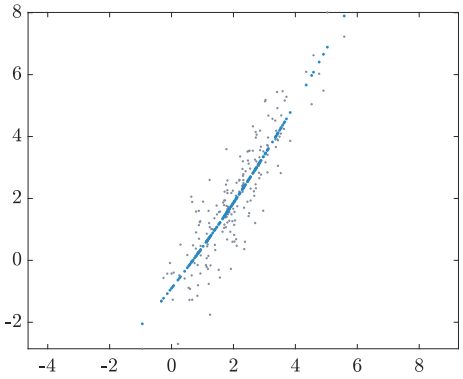

Figure 4: Non-linear (quadratic) basis function LS regression.

3 Connections

3.1 Singular Value Decomposition

3.2 Principal Component Analysis

4 Moore-Penrose pseudo-inverse

$$E = \sum_{j=1}^{k} \sum_{\substack{i=1 \\ x_i \in C_j}}^{N_j} ||x_i - c_j||^2 \tag{8}$$

If you have n people in a group and each shakes everyone else's hand only once, how many handshakes take place?

References

[1] G. Strang, *Introduction to Linear Algebra*, Fifth Edition, 2016

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006

[3] N. Kutz, *Data Driven Discovery of Dynamical Systems and PDEs*, an online lecture

[4] T. Hastie, R. Tibshirani, J. Friedman, *Elements of Statistical Learning, Data Mining, Inference, and Prediction*, Second Edition, 2008