# Applications of Linear Algebra in Linear Regression

**Mele P.**

*Computer Science, Data Science, Statistics, and Mathematics*
*Student at the University of Wisconsin-Madison*

----------***----------

**Abstract -** *In this research paper titled Applications of Linear Algebra in Linear Regression we will be exploring how the ways in which Linear Algebra contributes to this area of applied sciences.*

## 1 Least Squares and QR-Decomposition

QR-Decomposition allows for a matrix A to be expressed as a product of two matrices -- A = QR. These matrices are Q and R. Q is an orthogonal matrix whereas R is an invertible square upper-right triangular matrix. To Decompose A into a QR matrix pair, one would compute the following steps:

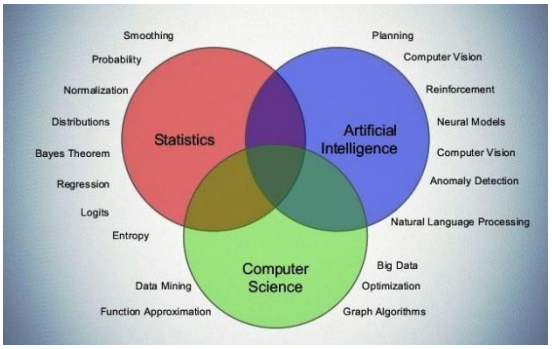
- Take columns of A and put them into an orthonormal set. Do this by doing the Gram-Schmidt process to get orthogonal vectors. After this, get the length of each vector and divide each matrix by their corresponding length. This will give one Q.
- To obtain R, there is a formula. This formula is this: (Q^T)A = R.
- After this, we've now obtained both Q and R for the QR-Decomposition
- To obtain (proj_w)b, we must first find x^(hat).
- x^(hat) = (R^-1)(Q^T)b
- (proj_w)b = (QR) x^(hat)

Least Squares allows for one to have a "best fit" line or curve for their data. It is typically used to solve inconsistent systems. To approach the "best fit", the goal is to seek a vector x^(hat) so that Ax^(hat) is as close to the solution vector b as much as possible. To solve a least-squares problem, the following steps must be made:

- Compute (proj_w)b by finding an orthonormal basis for W. To do this, we RREF A^T and transpose the nonzero rows. Next, do the Gram-Schmidt process on the previous to find the orthonormal basis for W.
- Another way is by using the following formula: x^(hat) = (((A^T)A)^-1) (A^T) b. or (A^T)Ax^(hat) = (A^T)b
- $\widehat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}.$
- (proj_w)b = A x^(hat)
- $\text{proj}_W \mathbf{b} = A\,\widehat{\mathbf{x}}$

### 1.1 Example of use in Machine Learning through Linear Regression

Linear Regression -- which is a Supervised Machine Learning model -- uses Least Squares and QR-Decomposition in its applications to create models to find the relationships between variables. Machine Learning (ML) is in the scope of Artificial Intelligence and Computer Science. It is essentially a field where scientists -- like statisticians, computer scientists, and mathematicians -- use a computer algorithm to help identify patterns, predict future outcomes, or optimize solutions -- all with little to no human intervention.




Supervised Learning is one way to implement Machine Learning Algorithms. Supervised Learning consists of three basic components. The first component is Labeled Data. Labeled Data consists of data that we have categorized based on similar characteristics. Direct feedback is second because after one trains their machine learning model, a test can be done for accuracy. Based on the results of this test, the method used to solve the machine learning problem could be changed to ensure that the program predicts to near accuracy. The third is Predict outcome/future because essentially, that is the goal of supervised machine learning algorithms.

### 1.2 How to approach Linear Regression using Linear Algebra

#### 1.2.1 QR-Decomposition Approach

To solve this using QR-Decomposition, we must first break down matrix X into matrices Q and R. We do this by following the steps outlined in the How-To-QR-Decomposition. From there, the formula would look like this in relation to the variables where **x^(hat) = (R^-1)(Q^T)b** and **(proj_w)b = (QR)x^(hat)**

Again, we proceed to plug in values directly to first solve for x^(hat), then next to solve for (proj_w)b. We yield the same results.

#### 1.2.1 Why does this approach work?

- Let's look at the formula from before:
- x^(hat) = (R^-1)(Q^T)b
- And this one as well:
- (A^T)Ax^(hat) = (A^T)b
- This formula originates from this sequence:
- A = QR
- {Q^T)A = (Q^T)QR= IR = R
- R = (Q^T)A
- (A^T)Ax^(hat) = (A^T)b
- (QR)^T(QR)x^(hat)=(QR)^Tb
- (R^T)(Q^T)QRx^(hat) = (R^T)(Q^T)b
- (R^T)Rx^(hat) = (R^T)(Q^T)b
- Rx^(hat)=(Q^T)b

From this sequence of steps, we can then put the data into basic algebra terms of systems of linear equations and solve for x^(hat). When we solve for x^hat, we obtain a vector containing x^(hat)1 and x^(hat)2. This can be put in similar algebra terms of x^(hat)2 = m(x^(hat)1) + b. This itself is the best-fit line. In the following paragraphs, we will go into how the best-fit line is composed of the idea of "squares" in which we attempt to minimize the sum of those squares.

#### 1.3.1 Least Squares Approach

To implement the least squares formula, the formula would look like this in relation to the variables:

**x^(hat) = (((A^T)A)^-1) (A^T) b**

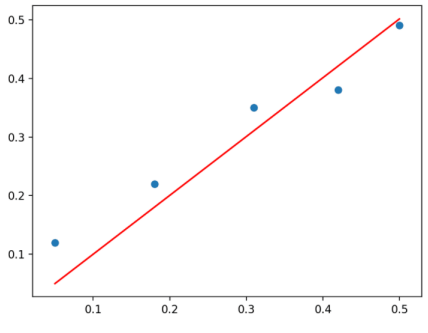
$$\hat{\mathbf{x}} = (A^T A)^{-1} A^T \mathbf{b}.$$

To find (proj_w)b, the formula would look like this in relation to the variables:

**(proj_w)b = A x^(hat)**

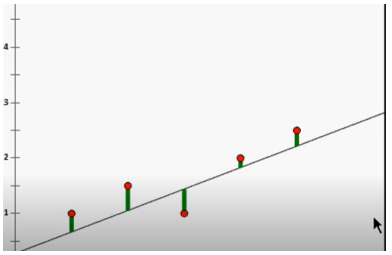
$$\text{proj}_W \mathbf{b} = A\hat{\mathbf{x}}$$

Assuming that matrix X and vector y data are given, we will first begin to find x^(hat) by plugging the values in directly. Next, we will use x^(hat) to find (proj_w)b by plugging those values in directly as well. As a result, we will yield (proj_w)b -- the best fit line that shows the most relation between the data points.

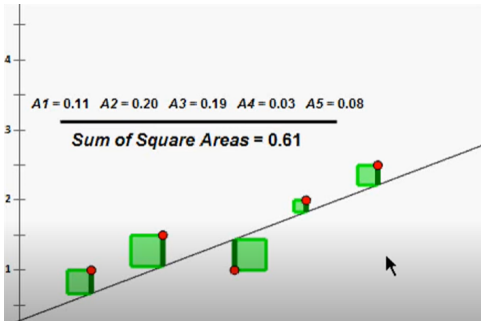



#### 1.3.2 Why does this approach work?

There is a term called residual lines. This refers to the distance between a given data point to the nearest point on the best-fit line. The most optimal best-fit line has a sum of its residuals equalling to zero -- where the distance above the line is positive and vice-versa for negative.



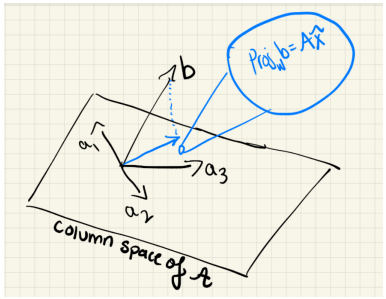
With the distances of the residual lines, one would square the values of them. The reason why "least squares" is called "least squares" is because one would want to create a sum of all the residual squared distances. The more the limit approaches 0, the better the line of best fit.




Similarly, this can be referred back to as the original vector b minus the projection of b on the column space of A -- commonly referred to as (proj_w)b throughout this paper.

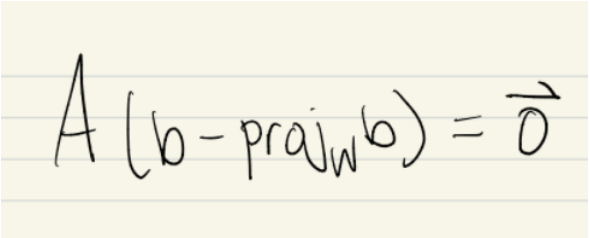
- **b - (proj_w)b = error = → 0**

We know Ax^(hat) = (proj_w)b. If we view the matrix A as a collection of columns [a1|a2|a3|...|an], then we can also think of a1,a2,..., an as vectors and view them out on a column space.




As you can see, there is a dotted line between the projection of b onto the column space of A and b itself.

- This is known as **b - (proj_w)b**. We can see that this is orthogonal to the column space of A. We can also see this interesting fact. Because **b - (proj_w)b** approaches 0, it ties back to the idea of a null space.
- A null space is the space of all possible vectors where if multiplied with a matrix A, it forms the 0 matrix and/or vector. This makes sense because with least squares, we want the minimal possible sum of all residual points -- in other words, to reach 0.
- In our case, it would visually look like this A(b-(proj_w)b) = 0(vector)




**2 Conclusion**

In conclusion, the purpose of using this math is to find a consistent linear correlation to non-linear data points. This can be applied in fields where prediction is necessary -- say, finance, computer science, etc. New data points will be added to collection and depending on how minimal the square of it is in relation to its residual and the best-fit line, professionals will be able to utilize it.

**REFERENCES**

[1] Ray, T.; Tanmoy Ray
I am a Career Adviser & Admission Consultant. Additionally; I am a Career Adviser & Admission Consultant. Additionally Beginner Guide Machine Learning, AI, IOT, NLP, deep learning https://www.stoodnt.com/blog/beginners-guide-to-machine-learning-artificial-intelligence-internet-of-things-iot-nlp-deep-learning-big-data-analytics-and-blockchain/ (accessed Sep 16, 2021).

[2] Supervised learning algorithm in Machine Learning https://techvidvan.com/tutorials/supervised-learning/ (accessed Sep 16, 2021).

[3] Raschka, S.; Mirjalili, V. Python Machine Learning Machine Learning and deep learning with python, scikit-learn, and tensorflow; Packt Publishing: Birmingham, 2018.

[4] Kolman, B.; Hill, D. R. Elementary linear algebra with applications; Pearson: New York, NY, 2018.