# Computational project 2

Goga Jincharadze

December 2024

## 1 Unconstrained least squares

Unconstrained least squares is a method used in regression analysis to estimate parameters by minimizing the sum of the squared residuals, in other words, minimizing the error.

Regression is used when we have a data set that we want to study and make predictions based on it, but I've used the normal equation method. For example, in my model, I've used data about various attributes of cars to make predictions about their gas mileage. My data consisted of features such as the year of the vehicle's release, it's highway miles per gallon, city miles per gallon, general miles per gallon, displacement, number of cylinders, model, drive, fuel type, etc. Some of these data, such as fuel type were non-numeric e.g diesel, gasoline, LPG, this types of columns were dropped from the data, since linear regression can only work on numeric data types. I've also dropped redundant data such as year of release, class, etc. In the end, I was left with a data set of 5 features: city mpg, highway mpg, displacement, number of cylinders and combination mpg.

Equation used:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

The normal equation:

$$\theta = (X^T X)^{-1} \cdot (X^T y)$$

### 1.1 Results of the multivariable linear regression

My data set had 547 rows of data, I've used 400 of it for training and the rest for testing and the results were excellent. Here is the graph:

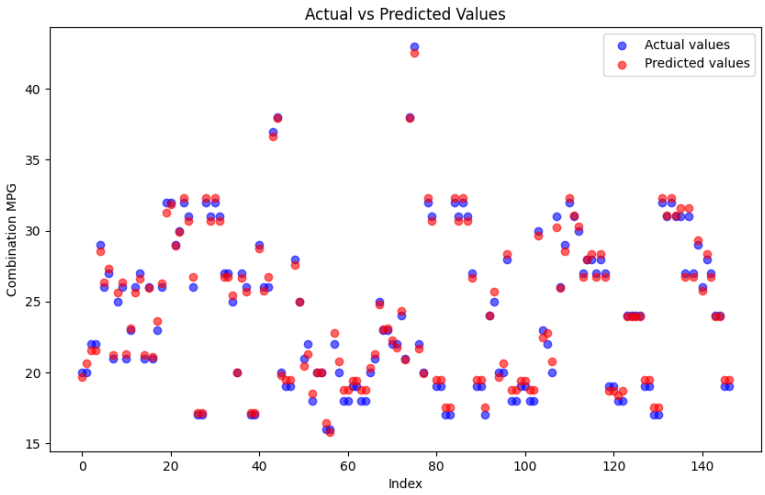


Figure 1: Graph of the result

## 1.2 Results of Classical Gram Schmidt

I simply factorized the problem int Q and R matrices using the classical approaches and then fed it to the regressor, the results were not good.

Why did this happen? Because of the numerical instability of the Classical Gram Schmidt method. Here are it's key issues:

1. Due to finite precision arithmetic of computers, during the iterative calculation of Q, it's columns may not be perfectly orthogonal, which leads to incorrect factorization.

2. Column dependence, if our X matrix has highly correlated columns, meaning that the different features of our data set are dependent on each other, the CGS will struggle to isolate the contribution of each column, this will lead to further errors.

3. Each step of projection and substraction has round off errors, so there is that also.

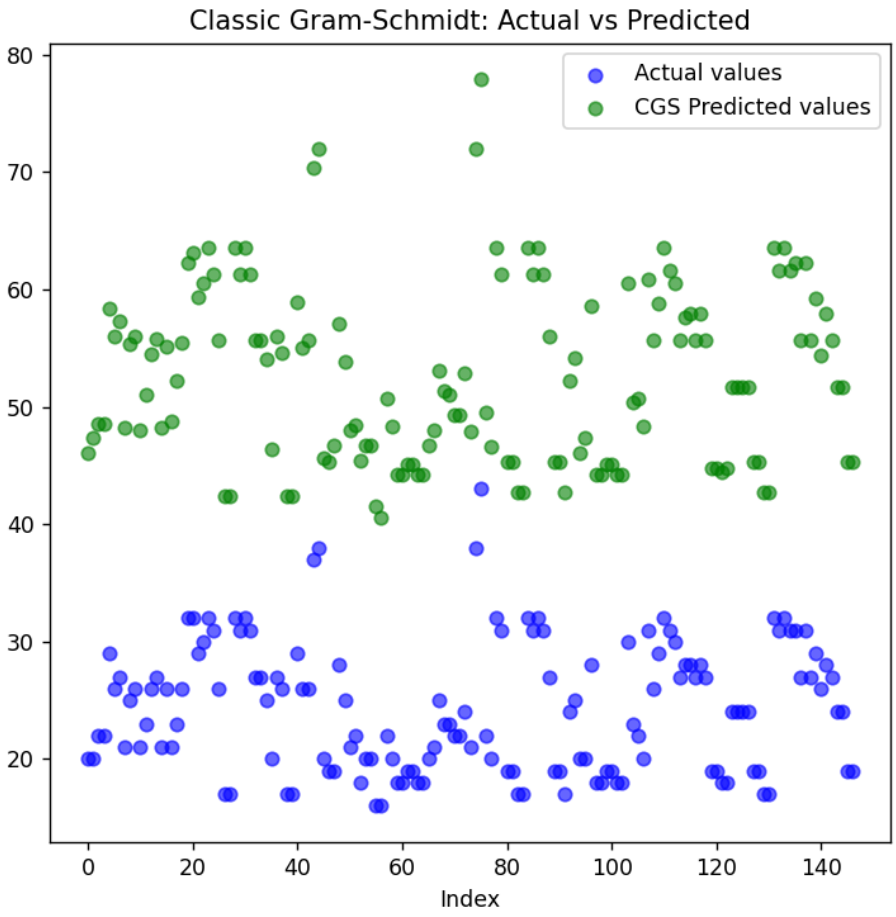


Figure 2: Graph of the result

### 1.3 Results of Modified Gram Schmidt

I factorized my data with using the modified approach and this time, the results were great.

Why did this actually worked unlike CGS? Because:

1.by updating vectors incrementally, preserves orthogonality better, getting more accurate Q and R matrices.

2. Dependent columns amplify the numerical instability in CGS because the projections and subtractions involve nearly parallel vectors. MGS handles this better by making the adjustments to the orthogonal basis incrementally.

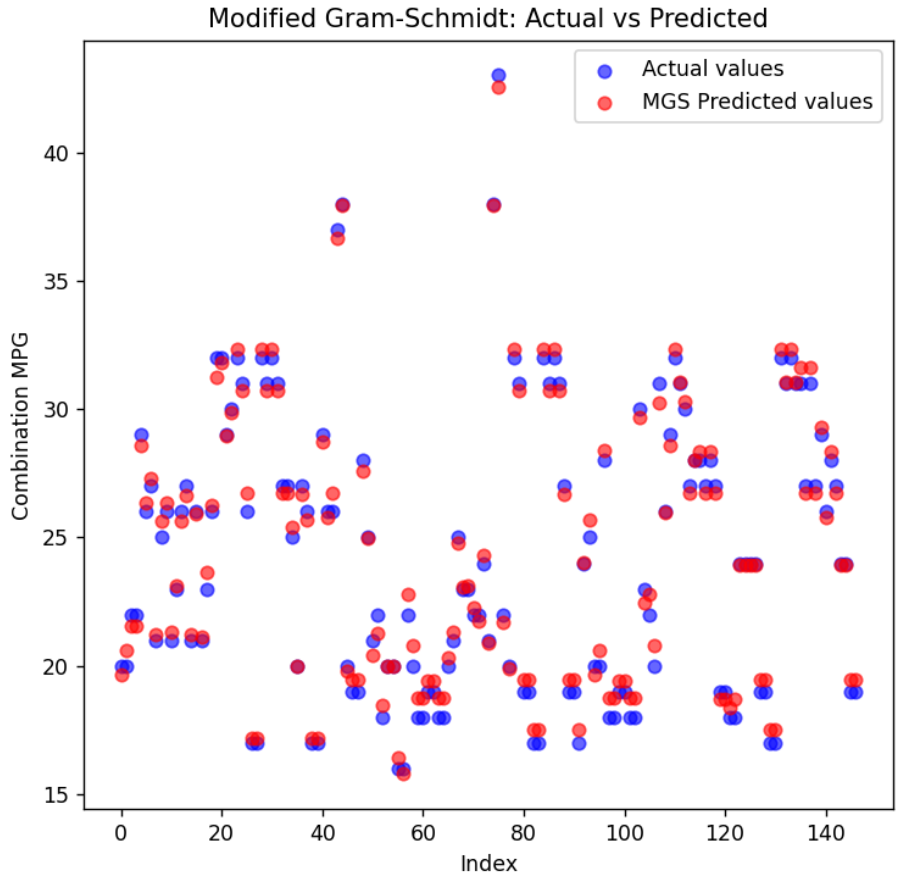


Figure 3: Graph of the result

3.When $X$ has correlated features, the condition number of $X^T X$ can be large. CGS struggles more in such cases, while MGS is more resilient to this instability.

## 2 Constrained least squares

Constrained least squares method is the same as the unconstrained one, but with a single key difference. In this case, we have a constraint, meaning something that must always be satisfied regardless of data. For example, I have applied this method on a data of pollutant concentrations collected across a specific geographic area. The stations that collect the data are not entirely accurate, they provide discrete data, but with this data and the use of constrained multivariate regression, we can estimate the true values of the concentrations of various pollutants. The constraint here is obvious, no pollutant concentration must be negative.

## 2.1 Results of the multivariate linear regression

My data set had approximately 9500 rows and 4 columns of various pollutants. Here is a legend:

PT08.S2(NMHC) - This represents the output of a sensor 2, that measures concentrations of non-methane hydrocarbons (NMHC)

NOx(GT) - The ground truth concentration of nitrogen oxides (NOx), a harmful pollutant produced primarily by combustion processes (e.g., vehicles, industrial activities).

PT08.S3(NOx) - The output of sensor 3, which is calibrated to detect nitrogen oxides (NOx).

NO2(GT) - The ground truth concentration of nitrogen dioxide (NO2), a specific component of NOx and a common pollutant harmful to health.

Here are the results:

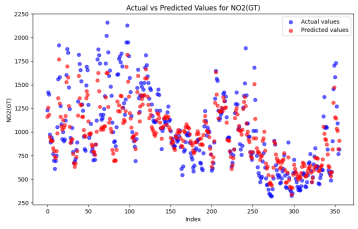


Figure 4: NO2(GT)

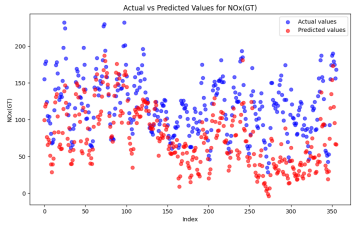


Figure 5: NOx(GT)

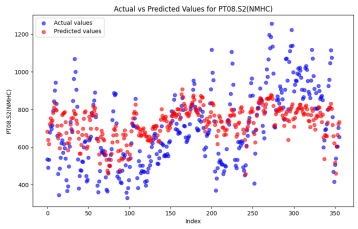


Figure 6: PT08.S2(NMHC)

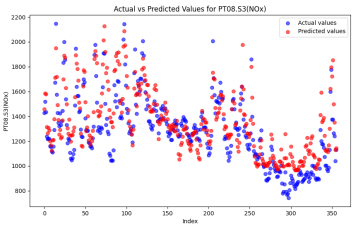


Figure 7: PT08.S3(NOx)

## 2.2 Results of Classical Gram Schmidt

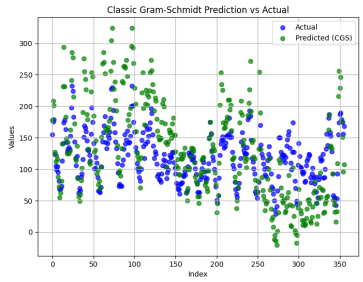


Figure 8: NO2(GT)

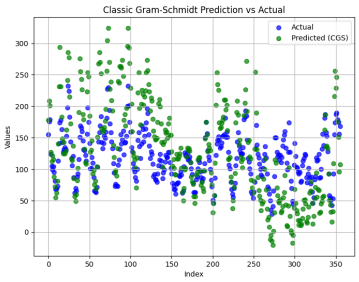


Figure 9: NOx(GT)

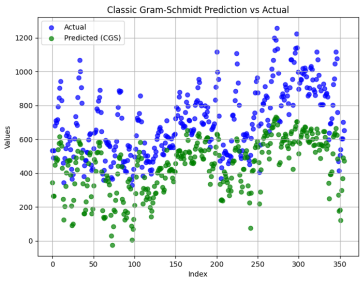


Figure 10: PT08.S2(NMHC)

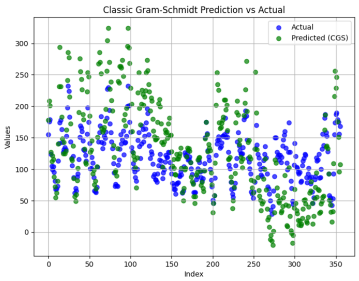


Figure 11: PT08.S3(NOx)

As you may have noticed, the predictions are accurate here.

## 2.3 Results of Modified Gram Schmidt

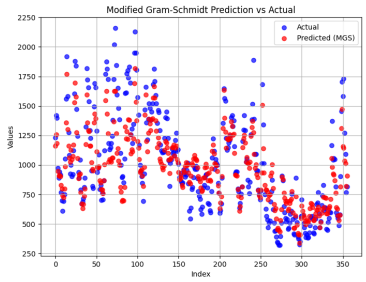


Figure 12: NO2(GT)

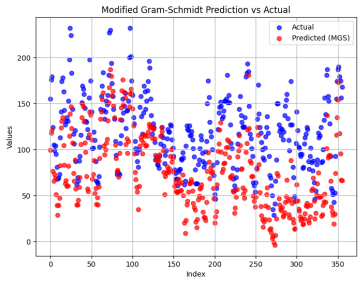


Figure 13: NOx(GT)

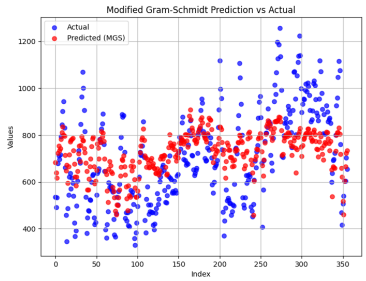


Figure 14: PT08.S2(NMHC)

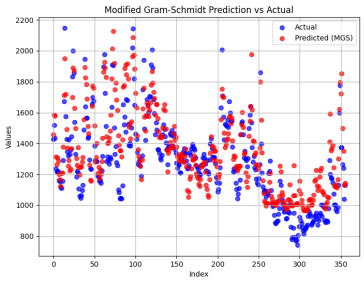


Figure 15: PT08.S3(NOx)

Modified version of GS definitely performed better here, the dependent columns were handled very well.

# 3 How to run the uploaded code

Every resource, code and data, is included in the uploaded folder, all you have to do is create a folder in which you will store the .csv data files and then correctly reference them in the code, then just run it, there are also comments in the code, which should help guide you through it.

By the way, you'll need numpy, pandas and matplot lib, respectively for, matrix operations, data normalization and reading, data visualization via graphs.