Predicting Concrete Compressive Strength using Linear Regression

Oswaldo David García Rodríguez

Tecnológico de Monterrey, Campus Querétaro

April 9, 2020

Abstract: This document presents the training process of a linear regression algorithm to estimate concrete compressive strength using a dataset of 1030 examples

INTRODUCCION

During construction, it is important to choose the most appropriate type of concrete that ensures the strength of the structures. The most used attribute is the concrete compressive strength, which depends on the type of mixture.

According to the article "Testing Compressive Strength of Concrete", this is measured using a compression-test machine that tries to break a cylindrical concrete sample. Then it is calculated the failure load divided by the cross-sectional area resisting the load. The result is represented in pound-force per square inch (psi) or megapascals (MPa).

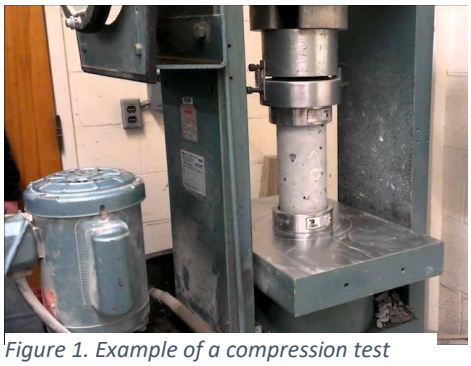

Figure 1. Example of a compression test machine

The use of a linear regression algorithm will try to make an estimation of the compressive strength of concrete given the concentrations for each material that makes up the mixture.

DATASET

It contains 1030 samples of concrete represented by 8 quantitative values and the compressive strength given in MPa.

The following attributes represents the components of the mixture, represented in kg per cubic meter:

1. Cement
2. Blast furnace slag
3. Fly ash
4. Water
5. Superplasticizer
6. Coarse aggregate
7. Fine aggregate

The last attribute is the age of the concrete, expressed in days.

LINEAR REGRESSION

This is a model which attempts to make a relationship between a set of inputs (x_i) and an output (y) using a linear equation.

The formula of a linear equation with multiple variables is represented as:

$$y = m_1x_1 + m_2x_2 + \cdots + m_nx_n + b$$

where n represents the number of attributes, m_i is the scope for each x_i, x_i is the attribute and b (bias) is a constant.

In linear regression, the hypothesis function is the way to make an approximation to the observed data (y) through calculating the scopes for all variables.

$$h_\theta(x) = \theta_0x_0 + \theta_1x_1 + \cdots + \theta_nx_n$$

where θ_i are the coefficients and represents the scopes of the x_i's. It's important to mention that θ_0 is the bias and x_0 it's always 1.

The first time, the values of coefficients can be random as they will be updated with each iteration to be closer to observed data. The difference between values obtained with hypothesis function and real values are calculated using the mean squared error (MSE) function:

$$J(\theta_0, \theta_1) = \frac{1}{m}\sum_{i=1}^{m}[h_0(x_i) - y_i]^2$$

where m is the number of samples, $h_0(x_i)$ is the hypothesis function applied to a sample and y_i is the true output of that sample.

SME won't affect the calculations of new coefficients, but it's useful to observe how the error changes with each iteration. If it is increasing, then it would mean the linear regression model should be modified.

To update the coefficients, each one changes using the cost function with respect to the corresponding coefficient multiplied with a learning rate α:

$$\theta_{j_{new}} = \theta_{j_{old}} - \alpha\frac{\delta}{\delta\vartheta_j}J(\theta_0, \theta_1)$$

$$= \theta_{j_{old}} - \frac{\alpha}{m}\sum_{i=1}^{m}[(h_\theta(x_i) - y_i)x_i]$$

The smaller the α, the smaller the change of coefficients. This means the algorithm will be slower, however, if α is increased, the model has a higher risk that the hypothesis function won't approach the desired accuracy.

DATA NORMALIZATION

It's common to find datasets with some variables that have values much smaller than others. In this case, for example, Superplasticizer varies from 0 to 20, while Coarse Aggregate minimum value is 850.

In cases like that, the smaller variables don't have a significant weight to the hypothesis function updating, so after a big number of coefficient updates (also called epochs), those values would be almost completely ignored.

In order to avoid this kind of problem, each set of attributes is updated using a scaling function:

$$scaled_x = \frac{current_x - average_x}{\sigma_x}$$

where σ_x is the standard deviation of the variable:

$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

This will result of all samples having values between -1 and 1, allowing all

variables to have the same impact on the hypothesis function.

MODEL TRAINING

For this algorithm, the learning rate α was defined as 0.001. Also, the number of samples used where 80% of the entire dataset. The rest of samples were used as test data.

Finally, the number of epochs defined was 10,000.

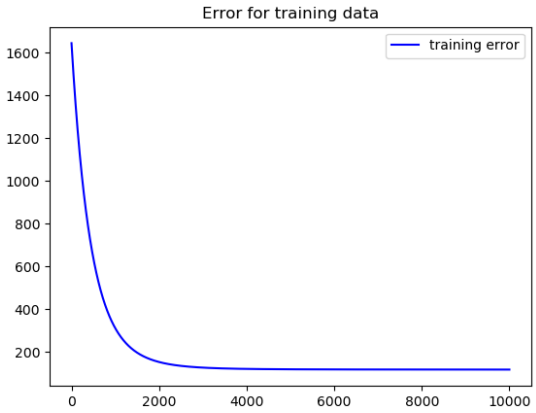

Figure 2. Error evolution of the model

It can be observed that, approximately since iteration 3,000, the error evolution started to be not significant.

For training data, SME was 118.5825 and accuracy of 77.7304%. For test data, SME was of 86.8542 and accuracy of 79.6704%.

REFERENCES

Jamal, H. (2017). *Procedure for Concrete Compression Test.* Retrieved from: https://www.aboutcivil.org/method-process-compression-test.html

NRMCA. (2014). *CIP 35 – Testing Compressive Strength of Concrete.* Retrieved from: https://www.nrmca.org/aboutconcrete/cips/35pr.pdf

UCI. (2007). *Concrete Compressive Strength Data Set*. Retrieved from: https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength