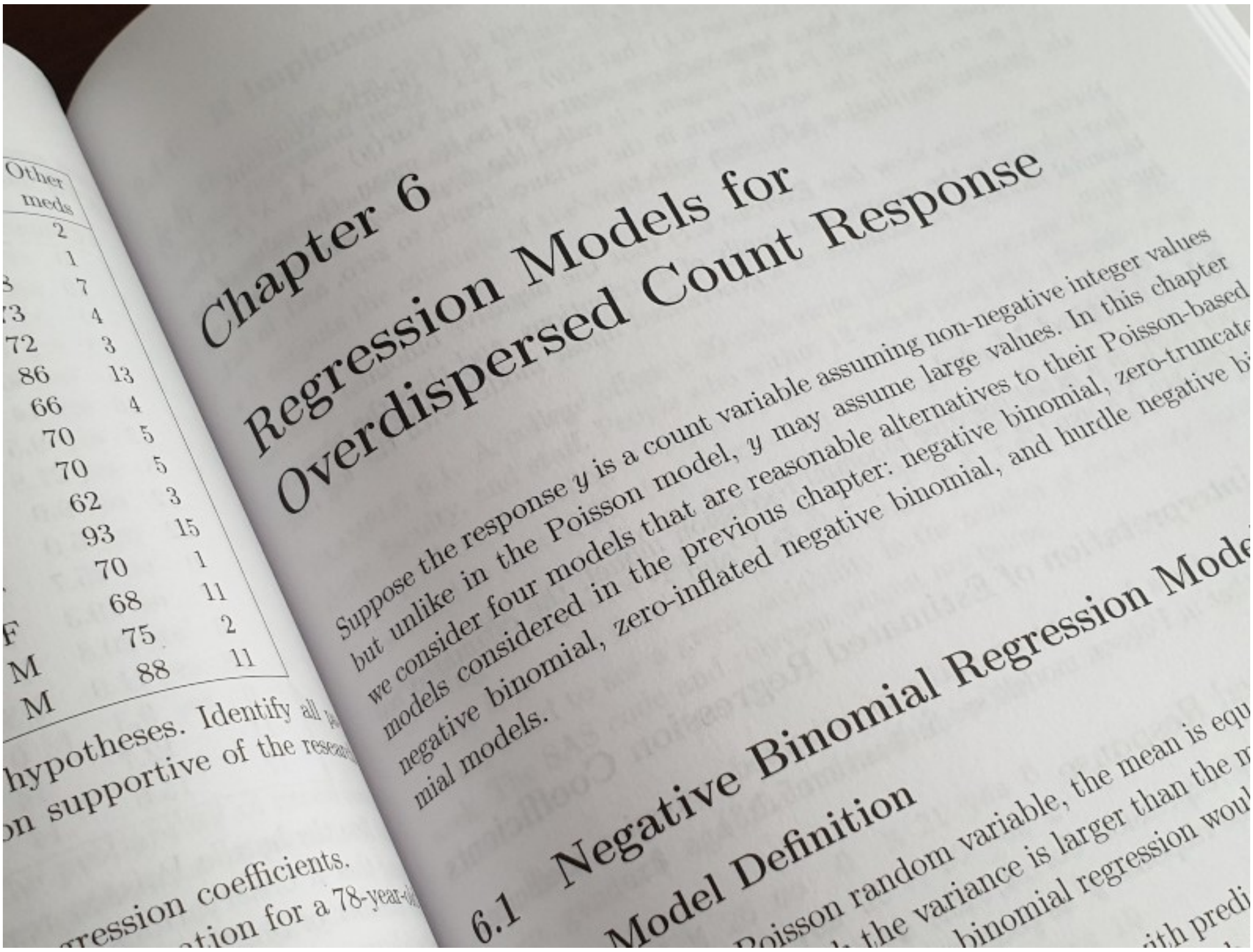
Linear Regression Test Data Error With A Simple Mathematical Formula

A concept every data scientist and machine learning researcher should remember



		Other meds
		2
		1
	3	7
	73	4
	72	3
	86	13
	66	4
	70	5
	70	5
	62	3
	93	15
	70	1
	68	11
F	75	2
M	88	11
M		

hypotheses. Identify all p
on supportive of the resea

ression coefficients.
ation for a 78-year-o

Chapter 6

Regression Models for Overdispersed Count Response

Suppose the response y is a count variable assuming non-negative integer values but unlike in the Poisson model, y may assume large values. In this chapter we consider four models that are reasonable alternatives to their Poisson-based models considered in the previous chapter: negative binomial, zero-truncate negative binomial, zero-inflated negative binomial, and hurdle negative bi mial models.

6.1 Negative Binomial Regression Mode

Model Definition

Poisson random variable, the mean is equ
the variance is larger than the m
binomial regression wou
ith predi

Photo by Enayet Raheem on Unsplash

1. Introduction

Linear regression is probably one of the most important concepts in statistical/machine learning because it is simple to understand, implement, and more importantly, many real situations can be modelled either as linear or can be reduced to linear by using appropriate mathematical transformations.

When we perform statistical/machine learning on a dataset(s), we split the data into training and test datasets. A very important quantity related to this splitting is the expectation value of the cost function on the test dataset(s), which is a quantity of great importance in machine learning. In a previous article, I showed how this expectation value is distributed among various quantities like the bias and variance, and in another article I showed how the bias-variance error is distributed by giving specific examples with Python. I would recommend having a look at these articles to understand the logical flow of many mathematical derivations that I present below.

As I have shown in my previous articles that have been mentioned above, the expectation value of the test data cost function is given by

$$E_{D,\epsilon}(C_{\text{test}}) = \text{Var}(\epsilon)+\text{Bias}^2[f(\boldsymbol{x},\bar{\boldsymbol{\theta}}_D)]+\text{Var}(f(\boldsymbol{x},\bar{\boldsymbol{\theta}}_D)) \qquad (1$$

Now suppose that one wants to perform a linear regression (simple or multivariate) and asks the question: What is the expected value of the total error of the test data? In this article, I will show you that the expression for the linear regression test data error is indeed very simple that every data scientist and machine learning researcher should always remember. In this article, I assume that the reader knows statistical theory, linear algebra, and calculus. The difficulty of this article is at an *intermediate to advanced level.*

2. Basic theory of linear regression

Here, I briefly outline the theory of multiple (or multivariate) linear regression that will be very useful in the next sections. Given a dataset *D = {y_i,* ***x****_(i)}* of *n* data points, where *i = {1,…, n}, y_i* are the components of the independent variable, and **x_***(i) is* the predictor vector corresponding to the independent variable *y_i*, the theory of multiple linear regressions assumes that there is a linear relationship of the type:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \boldsymbol{x}_i^\mathsf{T} \boldsymbol{\beta} + \varepsilon_i, \qquad (2)$$

where the vector ***β*** is the coefficient vector with *p+1* components, and the predictor vectors **x_***(i)* have *p+1* components. The symbol (T) in equation (2) represents the transpose of a vector or a matrix. Here, in accordance with my previous articles, *ε_i* represents the random error or noise variables that are assumed to be independent, identically distributed Gaussian variables with mean zero and variance σ^2.

Because we have n data points, in reality, equation (2) forms a system of linear equations that can be written in a more compact form

$$\boldsymbol{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \qquad (3)$$

where *X* the design matrix of shape with *n* rows and *p+1* columns, ***ε*** is the error column vector with *n* components, and ***y*** is the independent variable column vector with *n* components as well

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \qquad (4$$

The key point to remember is that equation (2) or (3) is our approximation to the true linear relation between the predictors and independent variables. It is an approximation because it includes the random error term. The goal is to find the vector $\boldsymbol{\beta}$ through a minimisation procedure, which in this article, I consider the Ordinary Least Square (OLS) procedure. This minimisation procedure requires that the Euclidean norm of the error term, must be minimum, namely $||\boldsymbol{\varepsilon}||^2 = ||X\boldsymbol{\beta}\text{-}\boldsymbol{y}||^2$ =minimum.

By making some simple calculations that involve by calculating the Euclidean norm of $||X\boldsymbol{\beta}\text{-}\boldsymbol{y}||^2$ and after minimising it by taking the partial derivative with respect to the vector $\boldsymbol{\beta}$**,** one gets:

$$\bar{\boldsymbol{\theta}}_D = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{y}, \qquad (5)$$

Equation (5) gives the vector $\boldsymbol{\theta}$ found by using the OLS minimization method. Here I am using the same notation as in my previous article to keep the logical flowing of the derivations that I present below. **One important thing about equation (5) is that it is valid only if the matrix product of the transpose of X with X is an invertible matrix**. This is usually true if $n>>p$, namely much more rows than columns. If the matrix rank, rank(X) = p, then the vector $\boldsymbol{\theta}$ in equation (5) is uniques, and if rank(X) < p (which is true when $p > n$), then $\boldsymbol{\theta}$ is not unique.

There are several methods to compute $\boldsymbol{\theta}$ that can be classified as direct methods and iterative methods. The direct methods include the Cholesky and QR factorisation methods and the iterative methods include the Krylov and Gradient Descendent methods. I do not discuss these methods in this article.

3. Linear regression test data error

In this section, I show you how to calculate the total error of linear regression on the test data and the result at the end may surprise you. As I showed in my previous article, equation (1) is one of the possibilities to express the test data averaged error through bias, variance, and noise. However, for the purpose of this article, it is better to use the final form of equation (5) of my previous article which I write as:

$$E_{D,\epsilon}(C_{\text{test}}) = \sigma_\epsilon^2 + \sum_{i=1}^{N} E_{D,\epsilon}\left(\left[f(\boldsymbol{x}_i) - f(\boldsymbol{x}_i;\bar{\boldsymbol{\theta}}_D)\right]^2\right), \qquad (6)$$

Calculation done by the author

Equation (6) is an equivalent form of equation (1) above in the text. One needs to pay attention that the sum in equation (6) is over the *test data points* and not training data, and the expectation value *E(.)*, is over the dataset *D* and error instance *ε***.** The true and learned functions appearing in equation (6), for the multivariate linear regression, are given by:

$$f(\boldsymbol{x}_i) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} \quad \text{(true function)}$$

$$\begin{aligned} &f(\boldsymbol{x}_i;\bar{\boldsymbol{\theta}}_D) = \boldsymbol{x}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}_D \quad \text{(learned function)} \\ &= \boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}(X\boldsymbol{\beta}+\boldsymbol{\epsilon}) \\ &= \boldsymbol{x}_i^{\mathrm{T}}\underbrace{(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}X}_{\boldsymbol{I}}\boldsymbol{\beta} + \boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{\epsilon} = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{\epsilon}, \qquad (7 \end{aligned}$$

Calculations done by the author for educational purposes

wherein equation (7) ***I*** is the identity matrix. Now I insert the true and learned functions in the second term in equation (6) and I get (I drop the summation symbol for the moment):

$$
\begin{aligned}
E_{D,\epsilon}\left(\left[f(\boldsymbol{x}_i) - f(\boldsymbol{x}_i;\bar{\boldsymbol{\theta}}_D)\right]^2\right) &= E_{D,\epsilon}\left(\left[\underbrace{\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{\epsilon}}_{\text{scalar quantity}}\right]^2\right) \\
&= E_{D,\epsilon}\left(\left[\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{\epsilon}\right]\left[\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\boldsymbol{\epsilon}\right]^{\mathrm{T}}\right) = E_D\left(\boldsymbol{x}_i^{\mathrm{T}}\underbrace{(X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\underbrace{E_\epsilon\left(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}}\right)}_{\sigma_\epsilon^2\boldsymbol{I}}X}_{\sigma_\epsilon^2\boldsymbol{I}}\left[\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}\right]^{\mathrm{T}}\right) \\
&= \sigma_\epsilon^2 E_D\left(\boldsymbol{x}_i^{\mathrm{T}}\left[\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}\right]^{\mathrm{T}}\right) = \sigma_\epsilon^2 E_D\left(\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}\boldsymbol{x}_i\right) \qquad (8)
\end{aligned}
$$

Calculations done by the author for educational purposes.

In deriving equation (8), I used different properties of **square matrices that have an inverse**. I do not show these properties here because I assume that the reader knows them.

Now I want you to pay very careful attention to the last term in equation (8). As you can see, in that expressions appears the expectation value over the training datasets, and the only variables that depend on the training dataset is the design matrix *X* and its transpose. If the training datasets are chosen randomly from a normal distribution of datasets, as is usually the case, then matrix *X* is a random matrix that depends on the training data.

When I derived the bias-variance error in my previous article (equation (1) above), I explicitly said that the expectation value over *D* was taken for the training datasets because due to the randomness in choosing these training datasets. However, by choosing random training datasets, in principle, this would also imply random test datasets **if** the data are split during the train-test procedure (for example 80%-20%) from the **same original dataset**. This implies, that the expectation value over *D* can be split as the expectation value of the training and test datasets:

$$E_D(\cdot) = E_{D_{\text{train}}}(\cdot)E_{D_{\text{test}}}(\cdot), \qquad (9)$$

The next step is to calculate the expectation values of the matrices in equation (9). Before doing the calculations explicitly, there are some important assumptions to be made. *These assumptions are that the training and test data predictor vector components are uncorrelated and normally distributed with mean zero, E(**x**) = **0**, and variance equal to one*. This can easily be done by standardising the random vector components in order to have mean zero and variance equal to one. Here I assume that the reader knows these procedures.

The next step is to look at the form of the *X* matrix in expression (4) and multiply it with its transpose. After the multiplication, one gets a square matrix with *(p+1)* rows and columns. The first element of this matrix on the top left is the number *n* and if one factorises this number outside the matrix, one is left with a matrix that has as elements the *arithmetic mean,* the *mean of the squares,* and *cross-correlation* of each predictor components. At this stage one invokes the theorem of large random numbers which states that for *n* very large or infinity, the *arithmetic mean* of random variables can be approximated with the mean (= *E(**x**)*), which in our case is zero by assumption. Also by using the fact that the vector components are uncorrelated and with variance one, the inverse of the product of *X* transposed with *X* is, for *n* very large, equal to 1/*n* times the *(p+1)* identity matrix. By using these arguments, I get:

$$E_D(\boldsymbol{x}_i^{\mathrm{T}}(X^{\mathrm{T}}X)^{-1}\boldsymbol{x}_i) = E_{D_{\text{test}}}\left(\boldsymbol{x}_i^{\mathrm{T}}\left[\underbrace{E_{D_{\text{train}}}(X^{\mathrm{T}}X)^{-1}}_{n^{-1}\boldsymbol{I}_{(p+1,p+1)}}\right]\boldsymbol{x}_i\right)$$

$$= n^{-1}E_{D_{\text{test}}}\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_i\right) = \frac{p+1}{n}, \qquad (10)$$

Calculations done by the author for educational purposes.

Now by using equation (10) in equation (8) and after replacing the result in equation (6) and summing over the test data points *N*, I get the following final result:

$$E_D(C_{\text{test}}) = \sigma_\epsilon^2 \left[1 + \frac{N(p+1)}{n} \right], \qquad (11)$$

Final expression for multiple linear regression test data error. Calculations done by the author for educational purposes.

4. Conclusions

In this article, I showed you how one can calculate the total test data error for multiple linear regression in machine learning. The final result is given in equation (11), and as I mentioned above, its expression is very simple and it depends on the training data number (*n*), test data number (*N*), and the number of predictors (*p*). A statistical/machine learning model "learns" well only in the case when *N(p+1)/n* is very close to zero, that might happen when *n>>N(p+1)*. One can play with the combination of these numbers to minimise as much as possible the total test data error.

It is important that you remind all assumptions made to derive equation (11). These assumptions are: the random error variables *ε_i* are *independent and identically distributed (i.i.d)* with mean zero and variance σ^2. The random training and test data predictors vector components are independent, normally distributed with zero mean and variance equal to one, and they are independent on *ε_i*. Another important assumption is that the training data predictor number (*n*) must be a very large number.

Clearly, the reader must keep also in mind the assumptions made in section 2 where I discussed the theory of multivariate linear regression.

If you liked my article, please share it with your friends that might be interested in this topic and cite/refer to my article in your research studies. Do not forget to subscribe for other related topics that will post in the future.

By Damian Ejlli on September 21, 2021.

Canonical link

Exported from Medium on October 8, 2021.