Fundaments of Machine learning for and with engineering applications

Reidar B. Bratvold, Enrico Riccardi[1]

Department of Energy Resources, University of Stavanger (UiS).[1]

Feb 26, 2024

A linear model

Considering a univariate case, we have:

$$q = f(x)$$

which relates the **independent variable** x to the **true dependent variable** q.

Assuming a linear model

$$q = \beta_0 + \beta_1 x$$

where β_i are the arbitrary selected coefficients.

Model set-up

For a given x we do not know the true response q, only the measurementa y_i for experiment i.

We have that:

$$y_i = q_i + \epsilon_i$$

NOTE: do not proceed if you do not fully understand this equation.

which is

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

Discussion point

Does ϵ_i matter? And why so?

Estimated model paramters

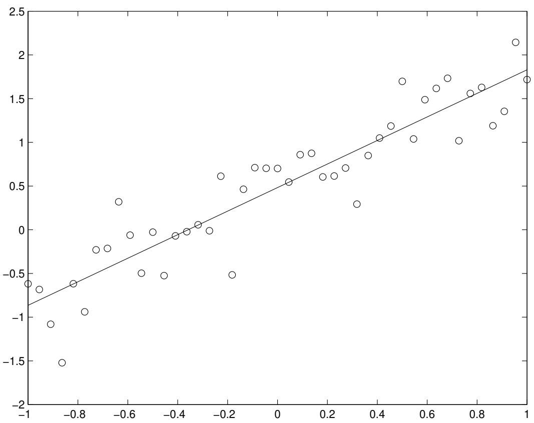
The model parameters β_0, β_1 are unknown, but they can be **estimated**. To distinguish estimates from true model parameters we call them b_0, b_1. These estimates are calculated such that the model

$$\hat{y} = b_0 + b_1 x$$

fits the n different experimental observations as well as possible.

Linear model example

We would like to find the TRUTH (What it it?)



Python Example

```
import numpy as np
import matplotlib.pyplot as plt

def generate_linear_data(n_random_points, noise=16):
    x = np.random.rand(n_random_points) * 10

    # Make 'perfect' data
    true_slope,  true_intercept = 2, 5
    y = true_slope * x + true_intercept

    # Add noise
    y += np.random.randn(n_random_points)*noise

    return x, y, true_slope, true_intercept

# Use the function to generate data
x, y, true_slope, true_intercept = generate_linear_data(
        n_random_points=166,
        noise=3)

# Plot all
plt.plot(x, true_slope*x + true_intercept,
         color='red', label='Truth Line')
plt.scatter(x, y, color='blue', label='Data Points')
plt.show()
```

Linear model(s)

Does a linear model mean only straight lines (or hyperplanes in general)?

That answer to this is *no*. different. In general for a model f to be defined linear, it has to be linear with respect to the unknown parameters $\beta_0, \cdots, \beta_n$. The general linear model is

$$q = \beta_0 + \beta_1 f_1(x_1) + \beta_2 f_2(x_2) + \cdots + \beta_n f_n(x_n)$$

where $f_i(x_i)$ may be non-linear functions. It is the **form** of the equation which makes it linear, i.e. that $f_i(x_i)$ does not depend on the parameters β_i.

Linear model(s)

Consider the following model example - is it linear?

$$q = \beta_0 + \beta_1 x_1^2 + \beta_2 x_2^{-1} + \beta_3 \log x_3$$

The answer is *yes* because by simple substitution it is possible to convert this formula into a linear form.

With $h_1 = x_1^2$, $h_2 = x_2^{-1}$ and $h_3 = \log x_3$, then we can formulate the new model:

$$q = \beta_0 + \beta_1 h_1 + \beta_2 h_2 + \beta_3 h_3$$

which is in the standard linear form.

Curvilinear models

A special class of linear models which we will investigate later are those which are expressed in terms of *polynomials* (here in only 1D):

$$q = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_m x^m = \sum_{i=0}^{n} \beta_i x^i$$

For n-D case there are a wide range of interaction terms and combinations, that can still be converted to standard linear form.

Such models are sometimes referred to as **curvilinear** instead of non-linear

Nonlinear models

But there are many other models that cannot be substituted to such a form. For instance:

$$q = \beta_0 + \log(x - \beta_1)$$

No substitution can transform this equation to the linear form.

That is the case for all the model that:

$$q \sim f(x, \beta)$$

Another example is:

$$f(x, \beta) = \frac{x\beta}{x + \beta}$$

Estimation of linear regression parameters

For the 1-dimensional problem, we have

$$\hat{y} = b_0 + b_1 x$$

where $\hat{y}$ is the estimated y-value from the approximate model that has been generated from a set of measurements (x_i, y_i). We aim to find the b_i parameters such that the regression line fits the observed data as well as possible.

This means we want to minimise the residuals

$$e_i = y_i - \hat{y}_i$$

Estimation of linear regression parameters

- Cannot sum e_i values since they might be positive and negative and thus cancel
- Could use e.g. $\sum_{i=1}^{n} |e_i|$, but is mathematically more difficult to handle
- Residual"smallness" measured by $\sum_{i=1}^{n} e_i^2$.

Thus, we find the linear regression coefficients by *minimising*

$$R = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

How?

Regression!

Let's recaps

Before to continue, let's make sure to have all the main elements clear:

1. Splitting the data in dependent and independent variables
2. Assumption of a linear model between them
3. Recognise the difference between the truth and the estimation
4. Aiming to *minimize* the residuals

Discussion

What happen when the sun of residual is 0 ?

What happens where the data is heavily correlated?

Regression

To minimize the sum of the square residuals, we can try to solve the following equations:

$$\begin{aligned} \frac{\partial R}{\partial b_0} &= 0 \\ \frac{\partial R}{\partial b_1} &= 0 \end{aligned}$$

where:

$$R = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 =$$

$$\sum_{i=1}^{n}(y_i - (b_0 + b_1 x_i))^2 = \sum_{i=1}^{n} u_i^2$$

Regression

Skipping the math (but you are more than welcome to try), here are the results:

$$\begin{aligned} b_0 &= \bar{y} - b_1\bar{x} \\ b_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \end{aligned}$$

Straightforward derivation becomes very cumbersome for multiple variables. Thus, a different approach must be used. Yet, it is important to understand that there is an analytical solution (even if not all the time).

Many variable equation

The general equation would look like:

$$\hat{y} = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n = \sum_{j=0}^{m} x_{ij} b_j$$

where we will have to solve all the $n+1$ equations (called the **normal equations**) of the form:

$$\frac{\partial R}{\partial b_j} = 0 \quad \forall j \in [0, n]$$

Is there a way for us to simplify this?

We can use vector and matrix algebra.

Regression via Matrix operation

Remember that

$$R = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

then we can define the vector $\mathbf{e}$:

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$$

thus

$$\boldsymbol{e}^T = [(y_1 - \hat{y}_1)\ (y_2 - \hat{y}_2) \cdots\ (y_N - \hat{y}_N)]$$

and can then write

$$R = \boldsymbol{e}^T \boldsymbol{e}$$

Regression via Matrix operation

Remember that

$$R = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

then we can define the vector $\mathbf{e}$:

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}}$$

thus

$$\boldsymbol{e}^T = [(y_1 - \hat{y}_1)\ (y_2 - \hat{y}_2) \cdots\ (y_N - \hat{y}_N)]$$

and can then write

$$R = \boldsymbol{e}^T \boldsymbol{e}$$

From the following equation:

$$\hat{y}_i = b_0 + \sum_{j=1}^{m} x_{ij} b_j = \sum_{j=0}^{m} x_{ij} b_j$$

where $x_{i0} = 1$
we make the matrix equation:

$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{b}$$

where the first column in $\mathbf{X}$ consists of ones only.

Residual

$$
\begin{aligned}
R &= \boldsymbol{e}^T\boldsymbol{e} \\
&= (\boldsymbol{y}-\hat{\boldsymbol{y}})^T(\boldsymbol{y}-\hat{\boldsymbol{y}}) \\
&= (\boldsymbol{y}-\boldsymbol{Xb})^T(\boldsymbol{y}-\boldsymbol{Xb}) \\
&= (\boldsymbol{y}^T-\boldsymbol{b}^T\boldsymbol{X}^T)(\boldsymbol{y}-\boldsymbol{Xb}) \\
&= \boldsymbol{y}^T\boldsymbol{y}-\boldsymbol{y}^T\boldsymbol{Xb}-\boldsymbol{b}^T\boldsymbol{X}^T\boldsymbol{y} \\
&+ \boldsymbol{b}^T\boldsymbol{X}^T\boldsymbol{Xb}
\end{aligned}
$$

All the parts of this equation are scalar values. This means e.g. that

$$\boldsymbol{y}^T\boldsymbol{Xb} = \boldsymbol{b}^T\boldsymbol{X}^T\boldsymbol{y}$$

This gives

$$R = \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{Xb} + \boldsymbol{b}^T\boldsymbol{X}^T\boldsymbol{Xb}$$

Residual

But how can we now compute $\frac{\partial R}{\partial b_j}$ more efficiently in matrix form?

Vector differentiation ! Let

$$y = \boldsymbol{a}^T\boldsymbol{x} = a_1x_1 + \cdots + a_nx_n$$

If

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \boldsymbol{a}$$

and $y = \boldsymbol{x}^T\boldsymbol{a}$, then:

$$\frac{\partial y}{\partial \boldsymbol{x}} = \boldsymbol{a}$$

General solution

In general, when $y = \boldsymbol{x}^T\boldsymbol{Ax}$, then

$$\frac{\partial y}{\partial \boldsymbol{x}} = 2\boldsymbol{Ax}$$

if **A** is symmetric
(check *Matrix calculus* for more properties)

We can use this to compute

$$\frac{\partial R}{\partial \boldsymbol{b}}$$

General solution

We have from above:

$$
\begin{aligned}
R &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{Xb} \\
&+ \boldsymbol{b}^T\boldsymbol{X}^T\boldsymbol{Xb}
\end{aligned}
$$

Vector differentiation gives

$$\frac{\partial R}{\partial \boldsymbol{b}} = 0 - 2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{Xb} = 0$$

Solving this for $\boldsymbol{b}$ we get:

$$
\begin{aligned}
\boldsymbol{X}^T\boldsymbol{Xb} &= \boldsymbol{X}^T\boldsymbol{y} \\
(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Xb} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \\
\boldsymbol{b} &= (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}
\end{aligned}
$$

Multiple linear regression

Previous equation make the solution of MLR rather obvious!

When we have a matrix of y-variables **Y**:

$$\boldsymbol{B} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

in the equation:

$$\boldsymbol{Y} = \boldsymbol{XB}.$$

These equations give us the **multiple linear regression** (MLR) solution.