Regression

Jordi Villà i Freixa

Universitat de Vic - Universitat Central de Catalunya
Study Abroad

jordi.villa@uvic.cat

course 2023-2024

Índex

1. Introduction and scope
2. Introduction
3. Simple and multiple linear regression
4. Regression framework
5. Analysis via linear models
6. Feature selection
7. Bibliography

Preliminary note

The material in these slides is strongly based on [1]. When other materials are used, they are cited accordingly.

Mathematical notation follows as good as it can a good practices proposal from the Beijing Academy of Artificial Intelligence.

What to expect?

In this session we will discuss:

- Learner
- Model assumptions
- Simple Linear Regression
- Multiple Linear Regression

Regression is a supervised learning method

The aim is to predict a quantitative response (output) variable y via a function $g(\mathbf{x})$ of an explanatory (input) vector $\mathbf{x} = [x_1, \dots, x_p]^T$.

In supervised learning[1]:

- The aim is to find a prediction function g that best guesses what the random output Y will be for a random input vector $\mathbf{x}$.
- The joint pdf $f(\mathbf{x}, y)$ of $\mathbf{X}$ and Y is unknown, but a training set $\tau = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ is available, which is thought of as the outcome of a random training set $\mathscr{T} = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$ of iid copies of $(\mathbf{X}, Y)$.

Learner

Let us assume a squarred-error loss function from now on, as the most common approach:

$$\mathrm{Loss}(y, \hat{y}) = (y - \hat{y})^2$$

Then, minimizing the risk function

$$\ell(g) = \mathbb{E}\mathrm{Loss}(Y, g(\mathbf{X})) = \frac{1}{n}\sum_{i=1}^{n}(y_i - g(\mathbf{x}_i))^2$$

gives us a way to evaluate the best function $g(\mathbf{x})$, which we will use for prediction and that we will call *learner*.

Simple linear regression

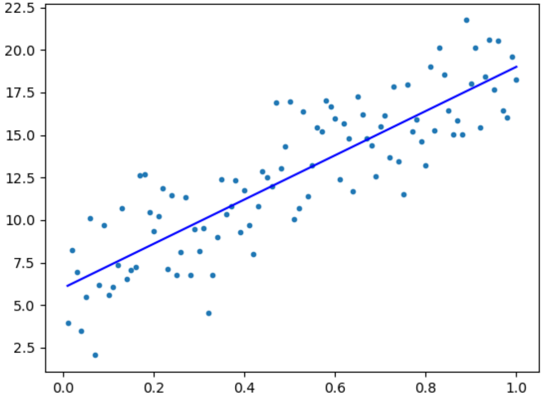


Figure 1: Let us assume some measurements $(x_i, y_i), \dots, (x_n, y_n)$ that lie approximately on a straight line.

Simple linear regression

In the simple linear regression, we assume that $\{x_i\}$ are fixed and that Y_i are random variables:

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \qquad i = 1, \dots, n$$

for some unknown parameters β_0, β_1 and $\{\varepsilon_i\}$ assumed independent with expectation value 0 and variance σ^2. We call $y = \beta_0 + \beta_1 x$ the *regression line*.

Multiple Linear Regression

Here we have a response Y that depends on a d-dimensional explanatory vector $\mathbf{x} = [x_1, \dots, x_d]^T$ via the relationship (for a single pair (x, Y)):

$$Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d + \varepsilon$$

And now the data lies approximately on a d-dimensional affine hyperplane

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \cdots + \beta_d x_d}_{g(\mathbf{x}|\beta)}$$

The function $g(\mathbf{x}|\beta)$ is linear with respect to β but not with respect to $\mathbf{x}$. In order to facilitate the calculations (and coding) we can augment the feature vector with the constant 1.

Multiple Linear Regression

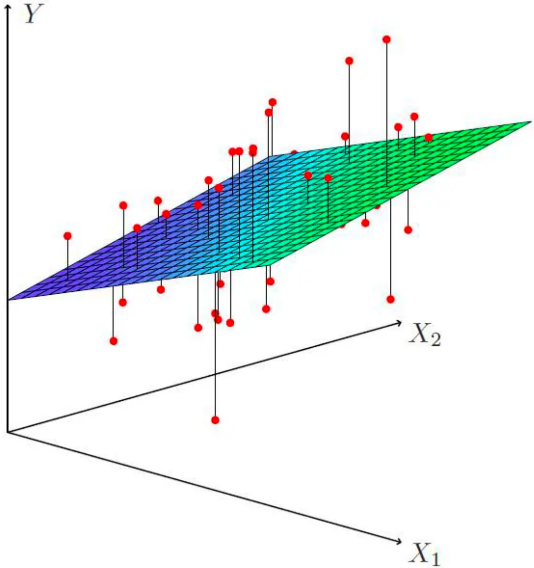
If instead that a single pair (x, Y), we consider the whole training set $\mathscr{T} = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$, by settling $\mathbf{Y} = [Y_1, \dots, Y_n]^T$ we can express the multiple linear regression modelfor the training set as:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

where $\;= [\varepsilon_1, \dots, \varepsilon_n]^T$ is a vector of iid copies of ε and $\mathbf{X}$ is the model matrix given by:

$$\mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{x}_1^T \\ 1 & \mathbf{x}_2^T \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^T \end{pmatrix}$$

Multiple linear regression



FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC-UCC

Parameter estimation

As we have seen, in the multiple linear regression model given by

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

$\mathbf{X}$ is assumed to be fixed, and only $\mathbf{Y}$ and ε are random.
Thus, the model contains two different parameters: β and σ^2 that need to be estimated from the training data τ. Given a linear prediction function $g(\mathbf{x}) = \mathbf{x}^T$, the squared error training loss is calculated as:

$$\ell_\tau(g) = \frac{1}{n}\left\|\mathbf{y} - \mathbf{X}\beta\right\|^2$$

The optimal learner g_τ minimizes this expression, leading to the expression that yields   through:

$$\mathbf{X}^T\mathbf{X}\beta = \mathbf{X}^T\mathbf{y}$$

Residuals

The traing loss can be then taken as an estimation of σ^2:

$$\hat{\sigma^2} = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\beta^2\|$$

The vector $\mathbf{e} = \mathbf{y} - \mathbf{X}\;$ is called the vector of residuals. $||\mathbf{e}||^2$ is called the residual sum of squares (RSS) and dividing RSS by $n-p$ gives an unbiassed estimate of σ^2, which we call the estimated residual squared error (RSE).

Data over- and underfitting

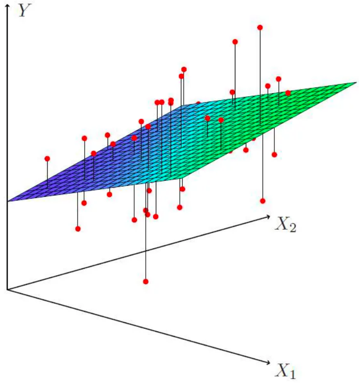


Figure 3: Including too fw features leads to large approximation error (underfitting) and including too many to large statistical error (overfitting).This leads to the need for feauture selection. Figure from Mathworks web site.

Dirk P. Kroese, Zdravko Botev, Thomas Taimre, and Radislav: Vaisman.
Data Science and Machine Learning: Mathematical and Statistical Methods.
Machine Learning & Pattern Recognition. Chapman & Hall/CRC, 2020.