

School of Electronic Engineering and Computer Science
Queen Mary University of London

CBU5201 Machine Learning

Supervised learning: Regression II

Dr Chao Liu

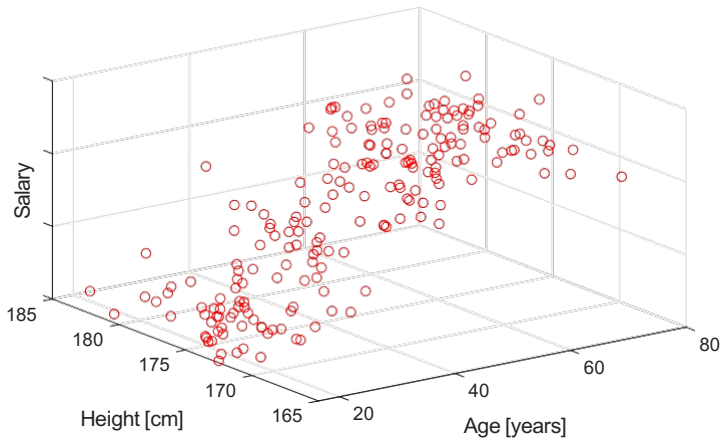
Sep 2023

Credit to Dr Jesús Requena Cañón



Multiple regression

In multiple regression there are **two or more predictors**. Given item i , we will denote each individual predictor as $x_{i,1}$, $x_{i,2}$, ... and $x_{i,K}$, where K is the number of predictors.



Multiple regression: Vector notation

Using vector notation, predictors can be packed together into a vector:

$$\mathbf{x}_i = [1, x_{i,1}, x_{i,2}, \dots, x_{i,K}]^T,$$

where $x_{i,k}$ denotes the k -th predictor of the i -th sample and K is the number of predictors. The constant 1 is prepended for convenience.

Using vector notation, multiple regression can then be expressed as

$$\hat{y}_i = f(\mathbf{x}_i)$$

Good news: the notation developed for simple regression can be easily translated to the multivariate scenario, no extra efforts required!

Multiple linear regression: Formulation

Linear models in multiple regression are simply the sum of a constant (or intercept) and each predictor multiplied by its own coefficient.

Multiple linear regression models can be expressed as:

$$f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i = w_0 + w_1 x_{i,1} + \cdots + w_K x_{i,K}$$

where $\mathbf{w} = [w_0, w_1, \dots, w_K]^T$ is the model's parameter vector.

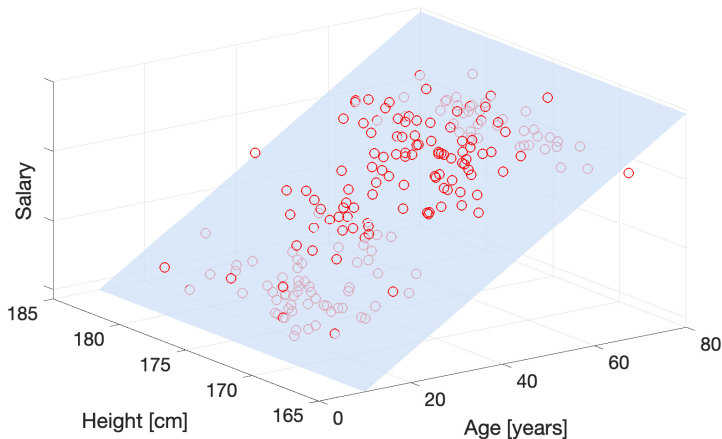
Note that we can use the same vector notation for simple linear regression models, by defining $\mathbf{w} = [w_0, w_1]^T$ and $\mathbf{x}_i = [1, x_i]^T$.

Multiple linear regression: Example

Use vector notation to represent a multiple linear regression model where the predictors are *age* and *height* and the label is *salary*.

Multiple linear regression: Solution visualisation

Multiple linear regression models are planes (or hyperplanes).



Multiple regression: More notation

In multiple linear regression, the **training dataset** can be represented by the **design matrix** \mathbf{X} :

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,K} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,K} \end{bmatrix}$$

and the **label vector** \mathbf{y} :

$$\mathbf{y} = [y_1, \dots, y_N]^T = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Multiple linear regression:

Example

Consider a dataset consisting of 4 samples described by three attributes, namely age, height and salary:

	Age [Years]	Height [cm]	Salary [GBP]
S_1	18	175	12000
S_2	37	180	68000
S_3	66	158	80000
S_4	25	168	45000

We decide to build a linear model that maps age and height to salary.

1. Use vector notation to represent the resulting linear regression model.
2. Obtain the design matrix \mathbf{X} and response vector \mathbf{y} .

The least squares solution

It can be shown that the **linear model** that minimises the metric E_{MSE} on a **training dataset** defined by a design matrix \mathbf{X} and a label vector \mathbf{y} , has the parameter vector:

$$\mathbf{w}_{best} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

This is an **exact** or **analytical solution** and is known as the **least squares** solution. It is valid for simple and multiple linear regression.

This solution can also be used for **polynomial models**, by treating the powers of the predictor as predictors themselves.

Note that the inverse matrix $(\mathbf{X}^T \mathbf{X})^{-1}$ exists when all the columns in \mathbf{X} are linearly independent.

Other models for regression

Linear and polynomial models are not the only options available. Other families of models that can be used include:

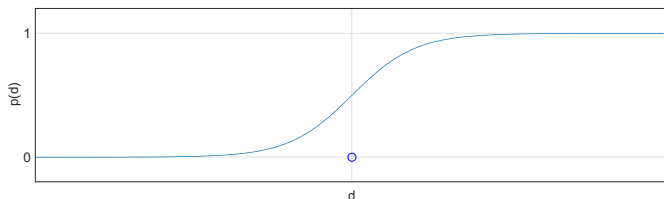
- Exponential
- Sinusoids
- Radial basis functions
- Splines
- And many more!

The mathematical formulation is identical and only the expression for $f(\cdot)$ changes.

Logistic model

In some problems, the label represents a **proportion** or a **probability**, i.e. a quantity between 0 and 1. Moreover, this quantity might increase as the predictor increases. In such cases, the logistic function can be useful.

The logistic function $p(d)$ is defined as $p(d) = \frac{e^d}{1+e^d} = \frac{1}{1+e^{-d}}$. By using $d = w_0 + w_1 x$, we can translate the logistic function and change its gradient.



Other quality metrics

In addition to the MSE, we can consider other quality metrics:

- **Root mean squared error.** Measures the sample standard deviation of the prediction error.

$$E_{RMSE} = \sqrt{\frac{1}{N} \sum e_i^2}$$

- **Mean absolute error.** Measures the average of the absolute prediction error.

$$E_{MAE} = \frac{1}{N} \sum |e_i|$$

- **R-squared.** Measures the proportion of the variance in the response that is predictable from the predictors.

$$E_R = 1 - \frac{\sum e_i^2}{\sum (y_i - \bar{y})^2}, \text{ where } \bar{y} = \frac{1}{N} \sum y_i$$

Agenda

Recap

Formulation of regression problems

Basic regression models

Flexibility, interpretability and generalisation

Summary

Flexibility

Models allow us to generate multiple shapes by tuning their parameters. We talk about the **degrees of freedom** or the **complexity** of a model to describe its ability to generate different shapes, i.e. its **flexibility**.

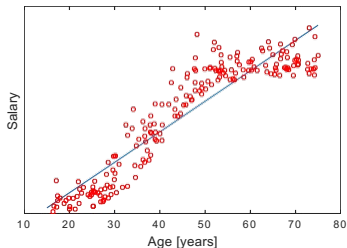
The degrees of freedom of a model are in general related to the number of parameters of the model:

- A linear model $y = w_0 + w_1x$ has two parameters and is inflexible, as it can only generate straight lines.
- A cubic model $y = w_0 + w_1x + w_2x^2 + w_3x^3$ has 4 parameters and is more flexible than a linear one.

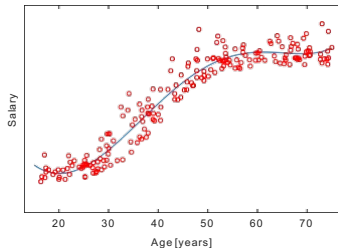
The flexibility of a model is related to its **interpretability** and **accuracy** and there is a trade-off between the two.

Interpretability

Model interpretability is crucial for us, as humans, to understand in a qualitative manner how a predictor is mapped to a label. Inflexible models produce solutions that are usually simpler and easier to interpret.



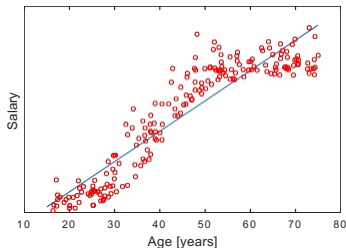
According to this linear model, the older you get, the more money you make



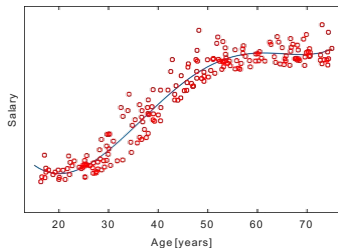
According to this polynomial model, our salary remains the same as teenagers, then increases between our 20s and 50s, then...

Quality on the training dataset

The quality of a model on a training dataset is also related to its flexibility. During training, the error produced by flexible models is in general lower.



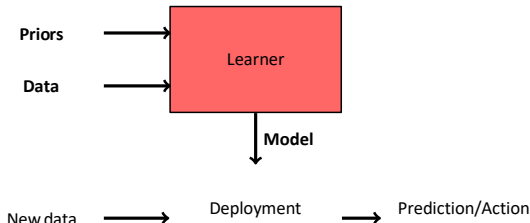
The **training error** of the best linear model is $E_{MSE} = 0.0983$



The **training error** of the best polynomial model is $E_{MSE} = 0.0379$

Generalisation

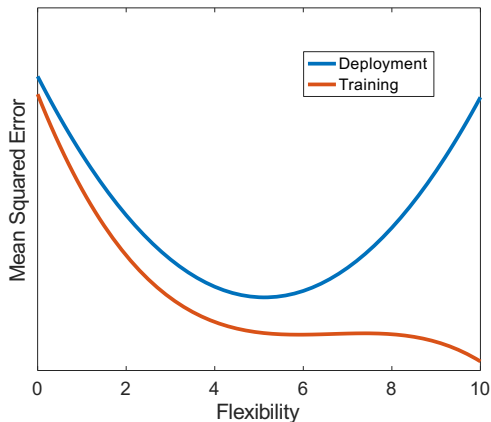
We have considered the **training MSE**, i.e. the quality of regression models on the **training dataset**.



Will our model work well during deployment, when presented with new data? **Generalisation** is the ability of our model to successfully translate what we was learnt during the learning stage to deployment.

Generalisation

In this figure, the red curve represents the **training MSE** of different models of increasing complexity, whereas the blue curve represents the **deployment MSE** for the same models. What's happening?

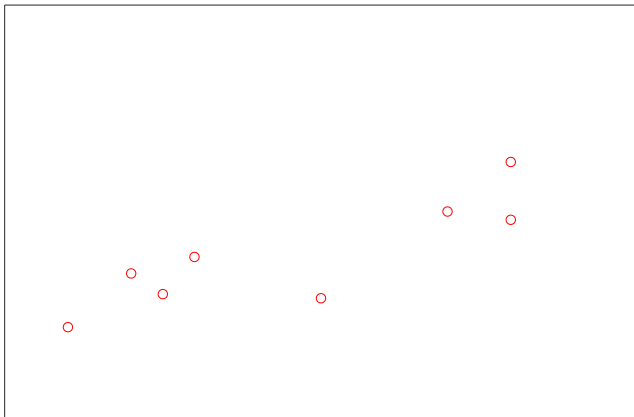


Underfitting and overfitting

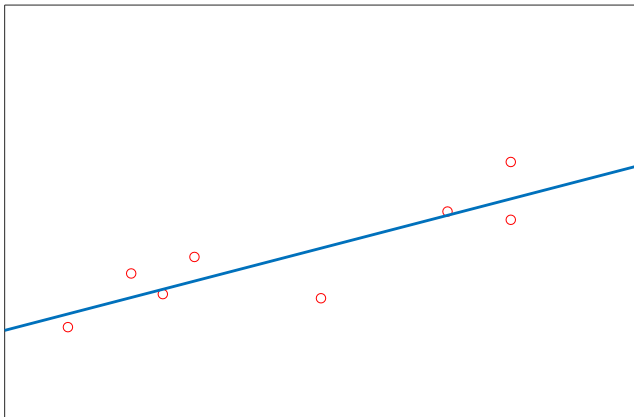
By comparing the performance of models during training and deployment, we can observe three different behaviours:

- **Underfitting:** Large training and deployment errors are produced. The model is unable to capture the **underlying pattern**. Rigid models lead to underfitting.
- **Overfitting:** Small errors are produced during training, large errors during deployment. The model is memorising **irrelevant details**. Too complex models and not enough data lead to overfitting.
- **Just right:** Low training and deployment errors. The model is capable of reproducing the **underlying pattern** and ignores **irrelevant details**.

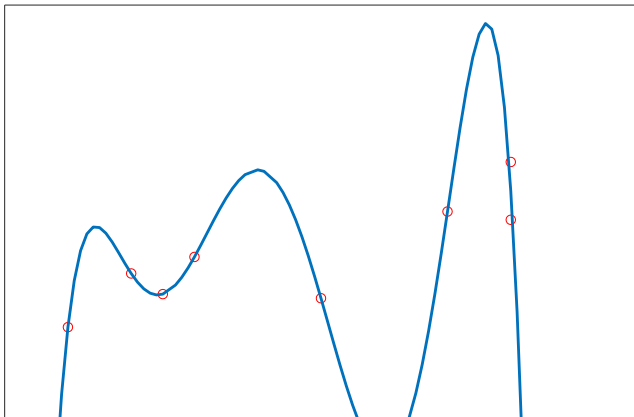
Underfitting and overfitting



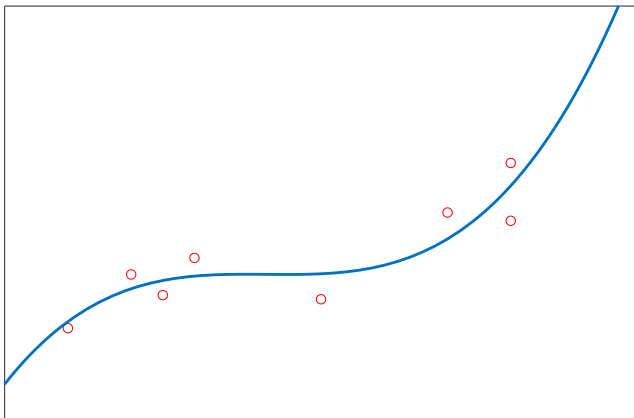
Underfitting



Overfitting

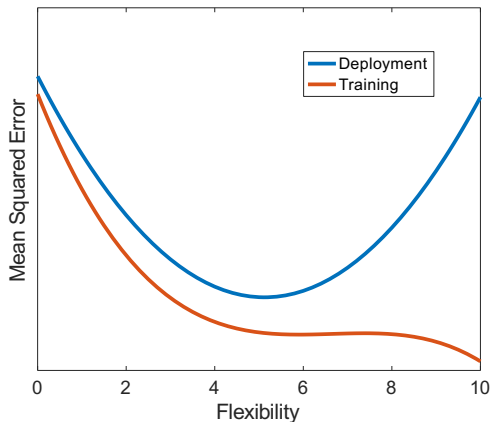


Just right



Underfitting and overfitting

Remember this: Generalisation can only be assessed by comparing training and deployment performance, not by just looking at how each model fits the training data.



Agenda

Recap

Formulation of regression problems

Basic regression models

Flexibility, interpretability and generalisation

Summary

Regression: Basic methodology

- Regression is a family of problems in machine learning, where we set out to find a model that **predicts a continuous label**.
- To build a model we use:
 - A **training dataset**,
 - a tunable **model**,
 - a **quality metric** and
 - an **optimisation** procedure.
- The final quality of a model has to be assessed during **deployment**.

Model generalisation

- Models have different degrees of **flexibility**. Complex models are flexible, simple models are rigid.
- A model **generalises** well when it can deal successfully with samples that it hasn't been exposed to during training.
- Three terms describe the ability of models to generalise:
 - **Underfitting**: unable to describe the underlying pattern
 - **Overfitting**: memorisation of irrelevant details
 - **Just right**: reflects underlying pattern and ignores irrelevant details

Final historical note

Wondering where the term *regression* comes from?

In the 19th century, Galton noticed that children of tall people tend to be taller than average – but not as tall as their parents. Galton called this *reversion* and later *regression towards mediocrity*.

This observation is nowadays called *regression to the mean*. You can read more about this curious fallacy in Kanehman's *Thinking, Fast and Slow*.