UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Machine Learning

Session 10 - T

Linear Models

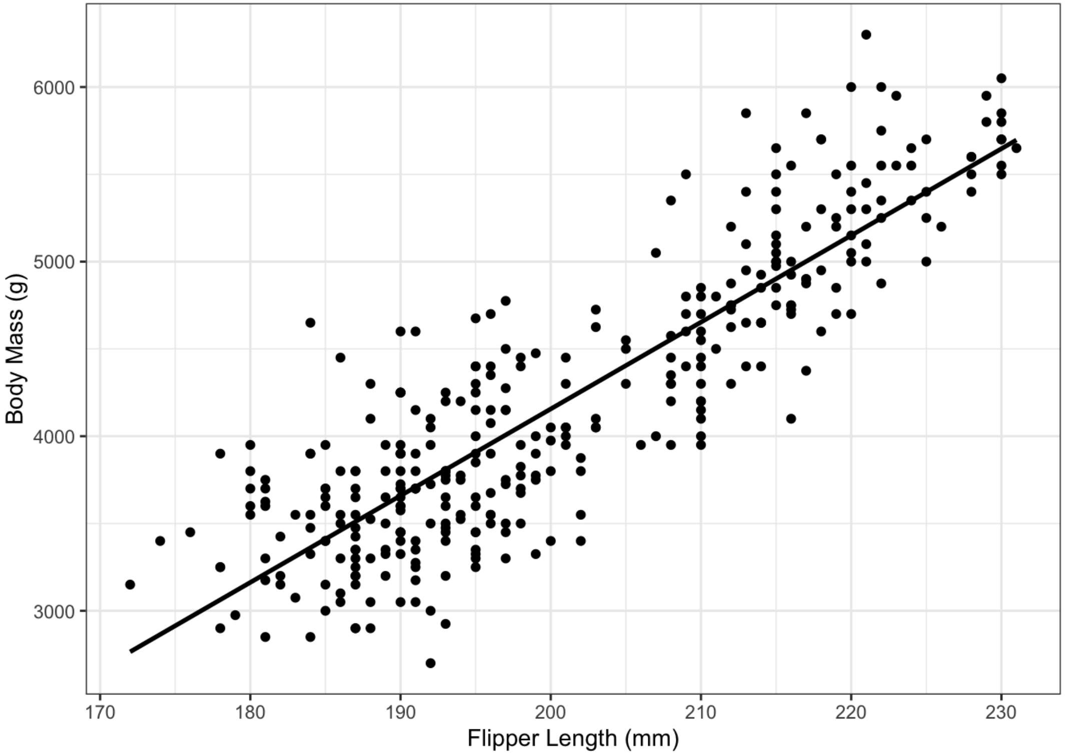
Degree in Applied Data Science

2024/2025

Linear Models

- Linear models are a class of machine learning algorithms that model the relationship between dependent and independent variables using **linear approximation**.

- **Key features:**
  - **Easy to interpret and implement**;
  - **Efficient** for large datasets due to linear computation;
  - Can be used for both **regression and classification tasks**.



Linear Models

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- A linear model is a **mathematical representation** of a relationship between a dependent variable y and one or more independent variables $x_1, x_2, ..., x_n$, where this **relationship is assumed to be linear**.

- Mathematically, it is represented as: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \epsilon$

Where y is the **dependent variable**,

$x_1, x_2, ..., x_n$ are the **independent variables**,

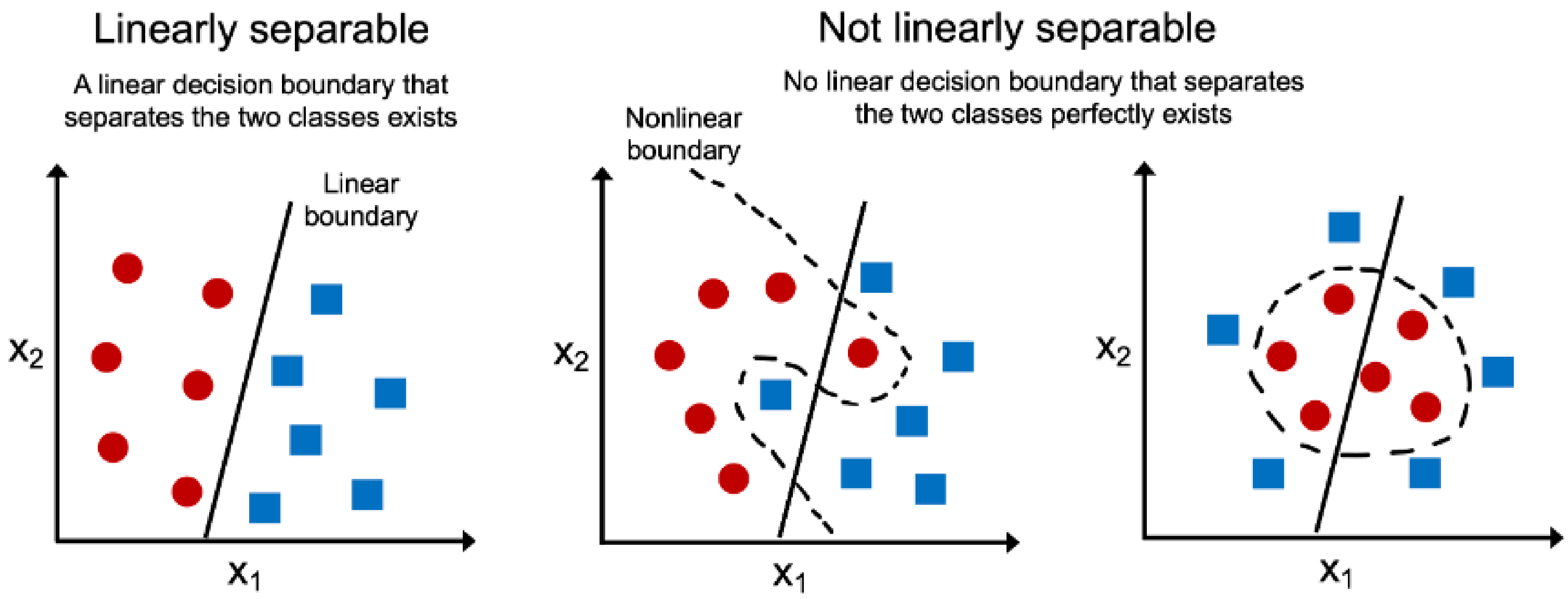
β_0 is the **intercept**,

$\beta_1, \beta_2, ..., \beta_n$ are the **coefficients** and

ϵ is the **error term**.

Non linear problems

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA



https://vitalflux.com/how-know-data-linear-non-linear/

- **Limitation:** In non linear cases, linear models may be insufficient...

Linear Models

- **Simple Linear Regression**
- **Multiple Linear Regression**
- **Polynomial Regression**
- **Logistic Regression**

Simple Linear Regression

- Simple linear regression models the relationship between **one independent variable and a dependent variable**. The equation for a simple linear regression is:

The intercept – a constant

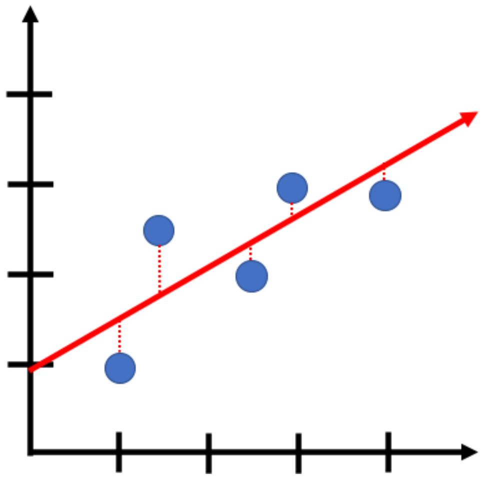
Independent Variable

$$y = \beta_0 + \beta_1 x + \epsilon$$

Error component

Dependent Variable

Coefficient\Weight\Slope - controls how much x contributes to y



Multiple Linear Regression

- Multiple linear regression extends simple linear regression to incorporate **multiple independent variables**. The equation for multiple linear regression is:

The intercept – a constant

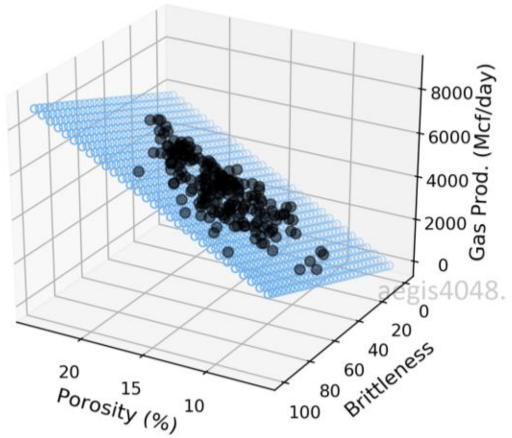
Independent Variables

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

Error component

Dependent Variable

Coefficients\Weights\Slopes - controls how much each x contributes to y



Polynomial Regression

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Polynomial regression involves fitting a curve to the data by introducing **polynomial terms**. The equation for polynomial regression is:

Polynomial terms

$$y = \beta_0 + \beta_1 x_1^1 + \beta_2 x_2^2 + \ldots + \beta_n x_n^n + \epsilon$$

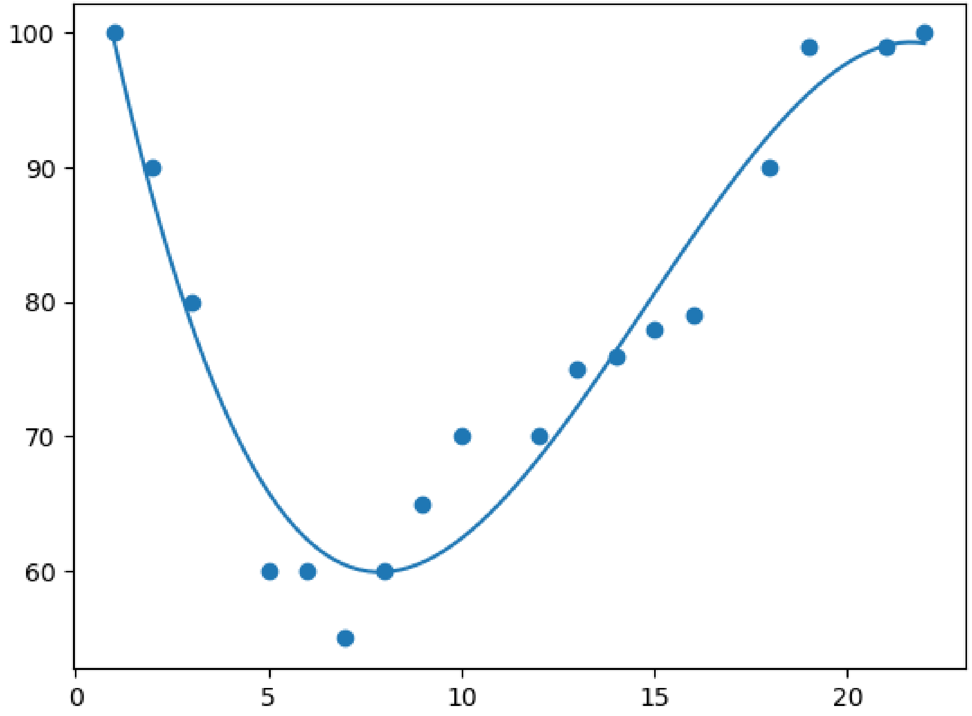
- NOTE: The term "linear" in linear regression refers to the **linearity of the coefficients**, not necessarily the linearity of the relationship between the independent and dependent variables.

- In polynomial regression, although the **relationship between the variables is modeled using a polynomial function** (which can appear curved), the estimation process is still linear with respect to the coefficients.

Polynomial Regression

Polynomial terms

$$y = \beta_0 + \beta_1 x_1^1 + \beta_2 x_2^2 + \ldots + \beta_n x_n^n + \epsilon$$



Linear Regression

- **Cost function:** Mean Squared Error (MSE)

$$J(\beta) = \frac{1}{2m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

Predicted value given by:

Real value

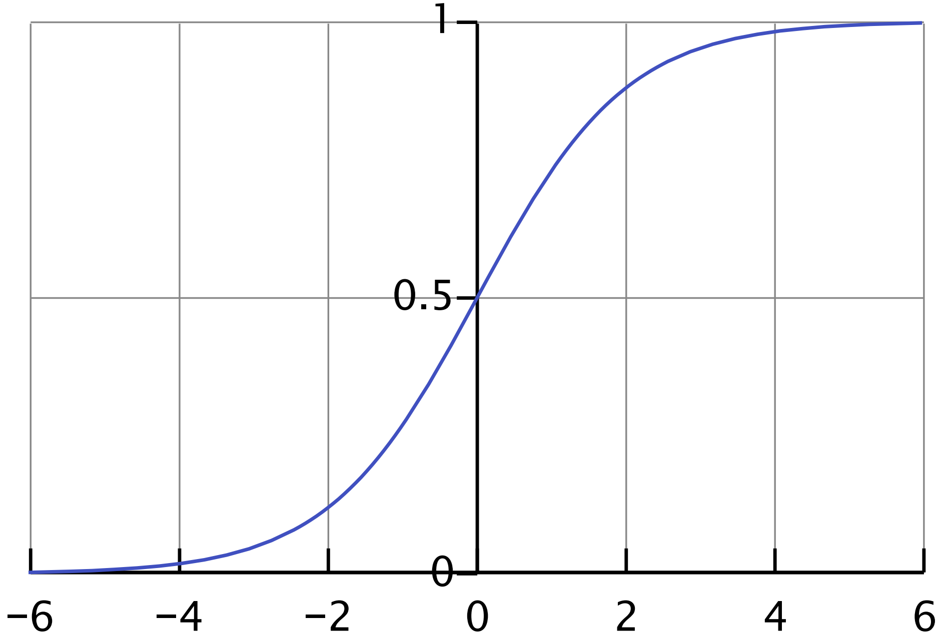
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

- **J** is a function of the model coefficients $\beta_1, \beta_1, \dots, \beta_n$
- Objective: identify the model coefficients that **minimize J**

Logistic Regression

- Logistic regression is used for **binary classification**.

- It models the **probability of the outcome belonging to a particular class** using the logistic function, also known as the **sigmoid function**.

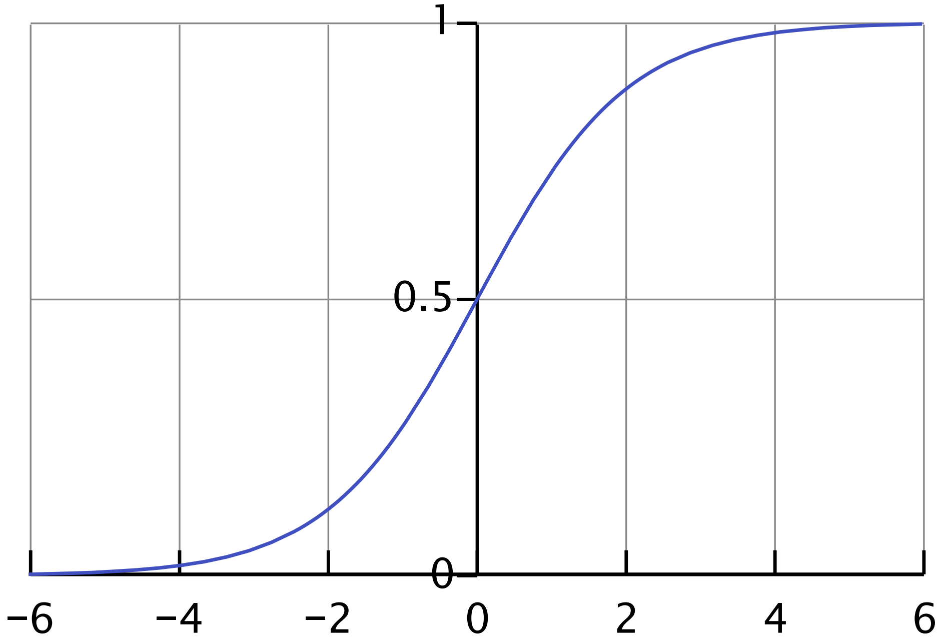


$$f(x) = \frac{1}{1 + e^{-(x)}}$$

Logistic Regression

- The predicted class is given by the **application of the sigmoid function to the linear regression output**.

- It gives us the **probability** of y (output) being 1 for the example x.



$$f(x) = \frac{1}{1 + e^{-(x)}}$$

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}}$$

Logistic Regression

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Train **separate "binary" models for each class**;
- Treat **other classes as a single entity** during training;
- **Each model estimates probability** of the example belonging to a class;
- Apply all models, **select class with highest predicted probability.**

Logistic Regression - Multiclass

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- **Cost function:**

$$J(\beta) = \begin{cases} -\log(\hat{y}) & \text{if} \quad y = 1 \\ -\log(1-\hat{y}) & \text{if} \quad y = 0 \end{cases}$$

- **Example:**
  - If true label y=1, prediction=0.8, error = -log(0.8) , low error
  - If true label y=0, prediction=0.8, error = -log(1 – 0.8), high error

- **Combining we get:** $J(\beta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(\hat{y}^{(i)}) - (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}}$$

Predicted value given by:

Real value

Parameter estimation (coefficients β)

- Parameter estimation consists in determining the **values of the coefficients** ($\beta_1, \beta_2, ..., \beta_n$) that **best fit the model** to the training data.

- This process involves finding the optimal parameters that **minimize a predefined loss function**.

- For linear models, we can use the **analytical method** of **Least Squares**, minimizing the error function **MSE**.

- Other **iterative methods** like **gradient descent** can also be used.

Linear Regression - Least Squares

- Algebric method that involves solving a **system of equations**:

$$\frac{\partial}{\partial \beta_j} J(\beta) = 0, j = 1, ..., n$$

$$\beta = (X^T X)^{-1} X^T y$$

Matrix version
X consists of the training data
+
first column with ones (to account for beta zero)

Linear\Logistic Regression – Gradient Descent

- Can only be applied if the **cost function is differentiable**.

- **Iterative method** that at each iteration changes the values of the parameters in order to minimize the error between predictions and true labels.

$$\beta_j := \beta_j - \alpha \frac{\partial}{\partial \beta_j} J(\beta)$$

Learning rate

The parameters are updated following:

$$\beta_j := \beta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x_j^{(i)}$$

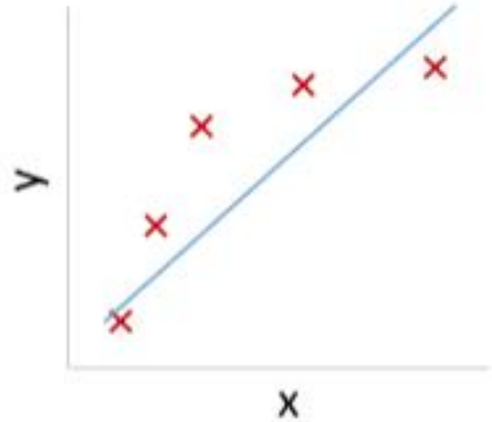
Least Squares vs Gradient Descent

- Least squares ensures **optimal solution**. Gradient descent may **not converge / get stuck** in local optima.

- Least squares is suitable and **computationally efficient** for small datasets;

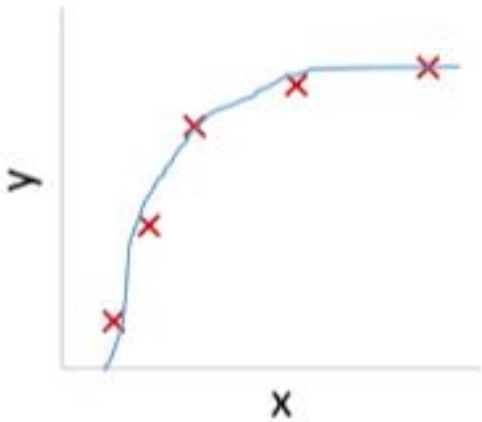
- Gradient descent is suitable for **large datasets** and can handle **non-linear models**.

Overfitting in Linear Models



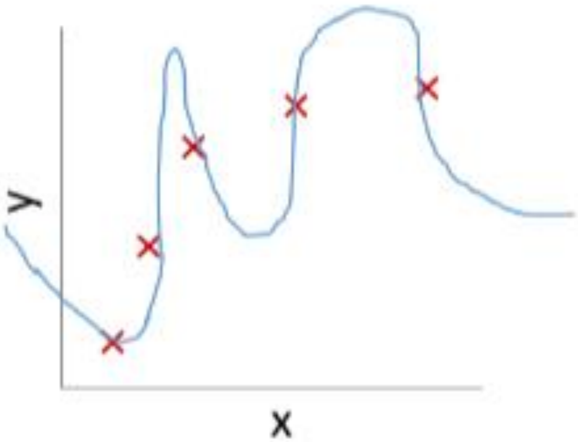
$$\theta_0 + \theta_1 x$$

Underfitting:
Insufficient model complexity



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

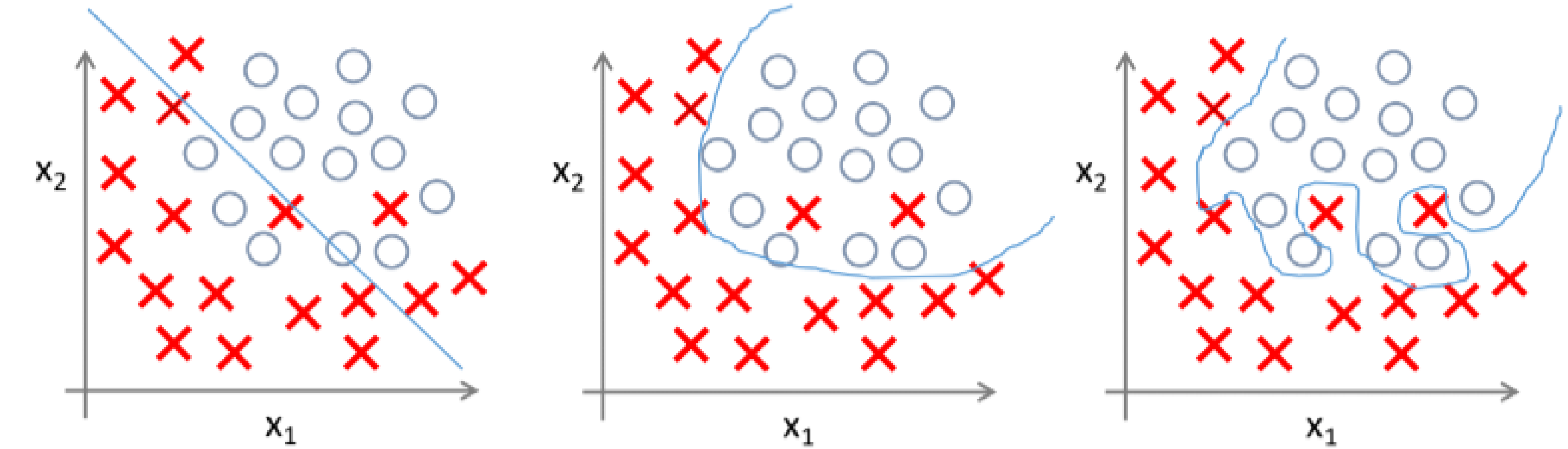
“Adequate” model complexity



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting:
Excessive complexity

Overfitting in Linear Models



$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

g = sigmoid function

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \ldots)$$

Underfitting:
Insufficient model complexity

"Adequate" model complexity

Overfitting:
Excessive complexity

Overfitting in Linear Models

- In linear models, like simple linear regression or multiple linear regression, **overfitting doesn't involve complex curves or surfaces bending to fit the data**.

- Instead, it involves the linear model **capturing noise or irrelevant fluctuations in the data**, which can happen when the model is too complex relative to the amount of data available.

- Linear models are prone to overfit in the cases where the **number of features is high** (specially when compared with the number of examples).

Overcoming Overfitting in Linear Models

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Reduce the number of features (coefficients) - feature selection.

- Regularization: Keep all features by try to reduce the magnitude of parameter values.

  - L1 regularization (**Lasso regression**)
  - L2 regularization (**Ridge refression**)
  - **Elastic nets** – use both L1 and L2 regularization

Ridge Regression

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Idea: **penalize high values of the parameters** (coefficients) in the cost function.

$$J(\beta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y}(i) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \beta_j^2\right]$$

Regularization parameter:
- higher values penalize parameter values more

If too high: risk of underfitting
If too low: risk of overfitting

Ridge Regression

- **Analytical method (least squares):**

$$\beta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \right)^{-1} X^T y$$

Ridge Regression

- **Gradient descent:**

$$\beta_0 := \beta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) x_0^{(i)}$$

$$\beta_j := \beta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

$$j = 1, \ldots, n$$

Term imposed by regularization
It is always < 1

Lasso Regression

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Idea: instead of penalizing the squared values of the parameters it penalizes the **absolute values**

$$J(\beta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y}(i) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \beta_j^2\right]$$ **Ridge**

$$J(\beta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(\hat{y}(i) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} |\beta_j|\right]$$ **Lasso**

- Everything else remains the same!

Regularization in Logistic Regression

- Cost function:

Ridge Regularization

$$J(\beta) = \left[-\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log(1 - \hat{y}^{(i)}) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \beta_j^2$$

$$J(\beta) = \left[-\frac{1}{m} \sum_{i=1}^{m} y^{(i)} \log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right) \log(1 - \hat{y}^{(i)}) \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} |\beta_j|$$

Lasso Regularization

Regularization in Logistic Regression

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

Gradient

$$\frac{\partial}{\partial \beta_0} J(\beta) \qquad \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})x_0^{(i)}$$

$$\frac{\partial}{\partial \beta_1} J(\beta) \qquad \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})x_1^{(i)} - \frac{\lambda}{m}\beta_1$$

$$\frac{\partial}{\partial \beta_2} J(\beta) \qquad \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})x_2^{(i)} - \frac{\lambda}{m}\beta_2$$

(...) (...)

$$\frac{\partial}{\partial \beta_j} J(\beta) \qquad \frac{1}{m}\sum_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})x_j^{(i)} - \frac{\lambda}{m}\beta_j$$

Resources

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Rencher, A. C., & Schaalje, G. B. (2007). Linear Models in Statistics (2nd ed.) [PDF]. doi:10.1002/9780470192610

- Pillonetto, G., Chen, T., Chiuso, A., De Nicolao, G., & Ljung, L. (2022). Regularization of Linear Regression Models. In Regularized System Identification (pp. 33–93). Springer International Publishing. https://doi.org/10.1007/978-3-030-95860-2_3

- Tran-Dinh, Q., & van Dijk, M. (2022). Gradient Descent-Type Methods: Background and Simple Unified Convergence Analysis (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2212.09413