in partnership with

ICR The Institute of Cancer Research

CANCER RESEARCH UK

The ROYAL MARSDEN NHS Foundation Trust

Linear algebra, calculus, least squares & logistic regression

02/05/2018

Sebastian Pölsterl, Priya Narayanan, Shan Raza, Igor Kozin

Plan

- 25/04: Introduction
- 02/05: Linear algebra, calculus, least squares and logistic regression
- 09/05: SVM; k-fold cross-validation and boosting
- 16/05: CNNs; Backprop; Representation Learning; Regularisation; SGD
- 23/05: Image classification using Deep Learning models; Keras, Tensorflow and TF-tensorboard

Please feel free to drop us an e-mail if you have any questions or suggestions.

Slides & Homework

You can find the presentation and homework exercises at:

https://github.com/ink1/dl-training

References

- Lecture series on linear algebra: https://www.khanacademy.org/math/linear-algebra
- 3Blue1Brown's series on linear algebra (https://youtu.be/kjBOesZCoqc) and calculus (https://youtu.be/WUvTyaaNkzM)
- Part I of the Deep Learning book: http://www.deeplearningbook.org/
- Linear algebra explained in four pages: https://minireference.com/blog/linear-algebra-tutorial/
- Andrew Ng Machine Learning course at Stanford: http://cs229.stanford.edu/syllabus.html
- The Matrix Cookbook: https://archive.org/details/imm3274

in partnership with

The Institute of Cancer Research

Cancer Research UK

The Royal Marsden NHS Foundation Trust

Linear Algebra

02/05/2018

Sebastian Pölsterl, Priya Narayanan, Shan Raza, Igor Kozin

Scalars

- A scalar is a single number: Integers, real numbers, rational numbers, …
- Usually denoted with italic font:

$$x, y, n$$

Vectors

- A vector is a 1-D array of scalars:

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$$

Usually denoted with lowercase, italic, bold font.

- A vector can contain real numbers, integers, …
- Example notation of type and size:

$$\boldsymbol{x} = \begin{pmatrix} 2 \\ \pi \\ -\sqrt{2} \end{pmatrix} \in \mathbb{R}^3$$

Matrices

- A matrix is a 2-D array of scalars:

A column

$$\boldsymbol{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{pmatrix}$$

A row

The main diagonal

- Usually denoted with uppercase, italic, bold font.
- $(\boldsymbol{X})_{i,j}$: First index denotes the row, the second index the column, e.g. $(\boldsymbol{X})_{4,2}$ denotes the scalar in the 4th row and 2nd column of matrix $\boldsymbol{X}$.
- Example notation of type and shape: $\boldsymbol{X} \in \mathbb{R}^{n \times m}$

Tensors

- A tensor is a an array of numbers of arbitrary dimensions.
- Scalars, vectors, and matrices are special cases of tensors.
  - A scalar is a zero-dimensional tensor
  - A vectors is a one-dimensional tensor
  - A matrix is a two-dimensional tensor

Matrix Transpose

- The transpose can be thought of as a mirror image across the main diagonal.

$$(\boldsymbol{X}^\top)_{i,j} = \boldsymbol{X}_{j,i}$$

$$\boldsymbol{X} = \begin{pmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \\ x_{3,1} & x_{3,2} \end{pmatrix} \Longrightarrow \boldsymbol{X}^\top = \begin{pmatrix} x_{1,1} & x_{2,1} & x_{3,1} \\ x_{1,2} & x_{2,2} & x_{3,2} \end{pmatrix}$$

$$((\boldsymbol{X})^\top)^\top = \boldsymbol{X}$$

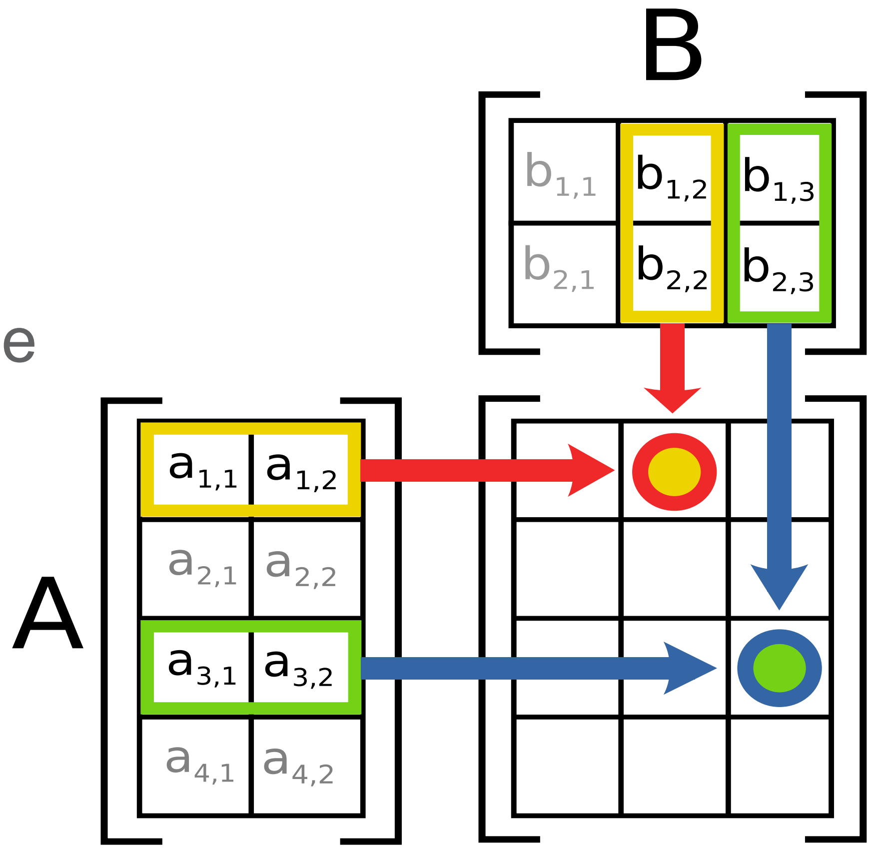
$$(\boldsymbol{X} + \boldsymbol{Y})^\top = \boldsymbol{X}^\top + \boldsymbol{Y}^\top$$

Matrix Product

- The product of two matrices $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times p}$ is the matrix $\boldsymbol{C} = \boldsymbol{AB} \in \mathbb{R}^{n \times p}$, where

$$(\boldsymbol{C})_{i,j} = \sum_{k=1}^{m} (\boldsymbol{A})_{i,k} (\boldsymbol{B})_{k,j}$$

- Note that the number of columns in ***A*** and must equal the number of rows in ***B***.



Matrix product – Properties

- $(\boldsymbol{AB})^\top = \boldsymbol{B}^\top \boldsymbol{A}^\top$
- The matrix product is …
  - distributive: $\boldsymbol{A}(\boldsymbol{B}+\boldsymbol{C}) = \boldsymbol{AB}+\boldsymbol{AC}$
  - associative: $(\mathbf{AB})\mathbf{C} = \boldsymbol{A}(\boldsymbol{BC})$
  - is in general **not** commutative: $\boldsymbol{AB} \neq \boldsymbol{BA}$

Inner product – Vector-Vector Product

- Special case of matrix multiplication between a row-vector ($1 \times n$) and a column-vector ($n \times 1$).
- Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, the inner product $\boldsymbol{x}^\top \boldsymbol{y}$ is a scalar $c \in \mathbb{R}$

$$\boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i = c$$

- Also called **dot product**.

Matrix-Vector product I

- The product of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times\mathrm{m}}$ and a vector $\boldsymbol{x} \in \mathbb{R}^{m}$ is a vector $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^{n}$.
- If we express **A** in terms of its rows, we can write the *i*-th entry of the product as the inner product of the *i*-th row of ***A*** and ***x***

$$\boldsymbol{y} = \begin{pmatrix} - & \boldsymbol{a}_1^\top & - \\ - & \boldsymbol{a}_2^\top & - \\ - & \vdots & - \\ - & \boldsymbol{a}_n^\top & - \end{pmatrix} \boldsymbol{x} = \begin{pmatrix} \boldsymbol{a}_1^\top \boldsymbol{x} \\ \boldsymbol{a}_2^\top \boldsymbol{x} \\ \vdots \\ \boldsymbol{a}_n^\top \boldsymbol{x} \end{pmatrix}$$

Matrix-Vector product II

- We can also multiply a matrix on the left by a row vector.
- The product of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times \mathrm{m}}$ and a vector $\boldsymbol{x} \in \mathbb{R}^{m}$ is a vector $\boldsymbol{y}^\top = \boldsymbol{x}^\top \boldsymbol{A} \in \mathbb{R}^{n}$.
- If we express **A** in terms of its <u>columns</u>, we can write the *i*-th entry of the product as the inner product of the *i*-th <u>column</u> of ***A*** and ***x***

$$\boldsymbol{y}^\top = \boldsymbol{x}^\top \begin{pmatrix} | & | & & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots & \boldsymbol{a}_m \\ | & | & & | \end{pmatrix} = \begin{pmatrix} \boldsymbol{x}^\top \boldsymbol{a}_1 & \boldsymbol{x}^\top \boldsymbol{a}_2 & \cdots & \boldsymbol{x}^\top \boldsymbol{a}_m \end{pmatrix}$$

Matrix-Matrix product

- The product of two matrices $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ and $\boldsymbol{B} \in \mathbb{R}^{m\times p}$ is the matrix $\boldsymbol{C} = \boldsymbol{AB} \in \mathbb{R}^{n\times p}$, where

$$(\boldsymbol{C})_{i,j} = \sum_{k=1}^{m} (\boldsymbol{A})_{i,k} (\boldsymbol{B})_{k,j}$$

- Thus, the entry $(\boldsymbol{C})_{i,j}$ is the inner product of the *i*-th row of ***A*** and the *j*-th column of ***B***.
- Alternatively, we can view matrix-matrix multiplication as a set of matrix-vector products by expressing ***B*** by its columns

$$\boldsymbol{AB} = \boldsymbol{A} \begin{pmatrix} | & | & & | \\ \boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots & \boldsymbol{b}_p \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \boldsymbol{Ab}_1 & \boldsymbol{Ab}_2 & \cdots & \boldsymbol{Ab}_p \\ | & | & & | \end{pmatrix}$$

Identity Matrix

- The **identity matrix**, denoted $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$, is a square matrix with ones on the diagonal and zeros everywhere else

$$\boldsymbol{I}_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

- It has the property that for any matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$,

$$\boldsymbol{A}\boldsymbol{I}_n = \boldsymbol{A} = \boldsymbol{I}_n\boldsymbol{A}$$

Diagonal matrix

- A **diagonal matrix** is a matrix where all non-diagonal elements are 0. This is typically denoted

$$D = \mathrm{diag}(d_1, d_2, \dots, d_n) = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & d_n \end{pmatrix}$$

Inverse of a matrix

- The inverse of a square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is denoted $\boldsymbol{A}^{-1}$, and is the unique matrix such that

$$\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}_n = \boldsymbol{A}\boldsymbol{A}^{-1}$$

- Matrix can't be inverted if …
  - ***A*** has more rows than columns,
  - ***A*** has more columns than rows,
  - ***A*** has redundant rows/columns ("linearly dependent", "low rank", "singular").

Inverse of a matrix – Properties

- The following properties assume that matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n}$ are invertible.

$$(\boldsymbol{A}^{-1})^{-1} = \boldsymbol{A}$$

$$(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$$

$$(\boldsymbol{A}^{-1})^{\top} = (\boldsymbol{A}^{\top})^{-1}$$

Linear System of Equations I

- Consider the following system of equations with the unknowns x_1 and x_2:

$$4x_1 - 5x_2 = -13$$
$$-2x_1 + 3x_2 = 9$$

- In matrix notation, we can write the system more compactly as $\boldsymbol{Ax} = \boldsymbol{b}$

$$\begin{pmatrix} 4 & -5 \\ -2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -13 \\ 9 \end{pmatrix}$$

Linear System of Equations I

- Consider the following system of equations

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$\begin{pmatrix} 4 & -5 \\ -2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -13 \\ 9 \end{pmatrix}$$

- Multiply by $\boldsymbol{A}^{-1}$ on both sides to obtain

$$\boldsymbol{A}^{-1}\boldsymbol{Ax} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

$$\boldsymbol{I}_2\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Linear System of Equations II

- A linear system of equations can have
  - No solution
  - Many solutions
  - Exactly one solution

Vector Norm

- A norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ denoted $||\boldsymbol{x}|| \in \mathbb{R}$ is a measure of the "length" of a vector.
- A norm must satisfy four properties:

1. Non-negativity: $||\boldsymbol{x}|| \geq 0$
2. Definiteness: $||\boldsymbol{x}|| = 0 \iff \boldsymbol{x} = \boldsymbol{0}$
3. Homogeneity: $\forall \alpha \in \mathbb{R}, ||\boldsymbol{\alpha x}|| = |\alpha|||\boldsymbol{x}||$
4. Triangle inequality: $||\boldsymbol{x} + \boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$

Common Vector Norms

- ℓ_1-norm

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

- ℓ_2-norm

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

- Max/Infinite-norm

$$\|\boldsymbol{x}\|_\infty = \max_i |x_i|$$

Special Matrices and Vectors

- The vector $\boldsymbol{x} \in \mathbb{R}^n$ is a **unit vector** if

$$\|\boldsymbol{x}\|_2 = 1$$

- The square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is **symmetric** if

$$\boldsymbol{A} = \boldsymbol{A}^\top$$

- The square matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is **orthogonal** if

$$\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}_n$$

$$\boldsymbol{A}^{-1} = \boldsymbol{A}^\top$$

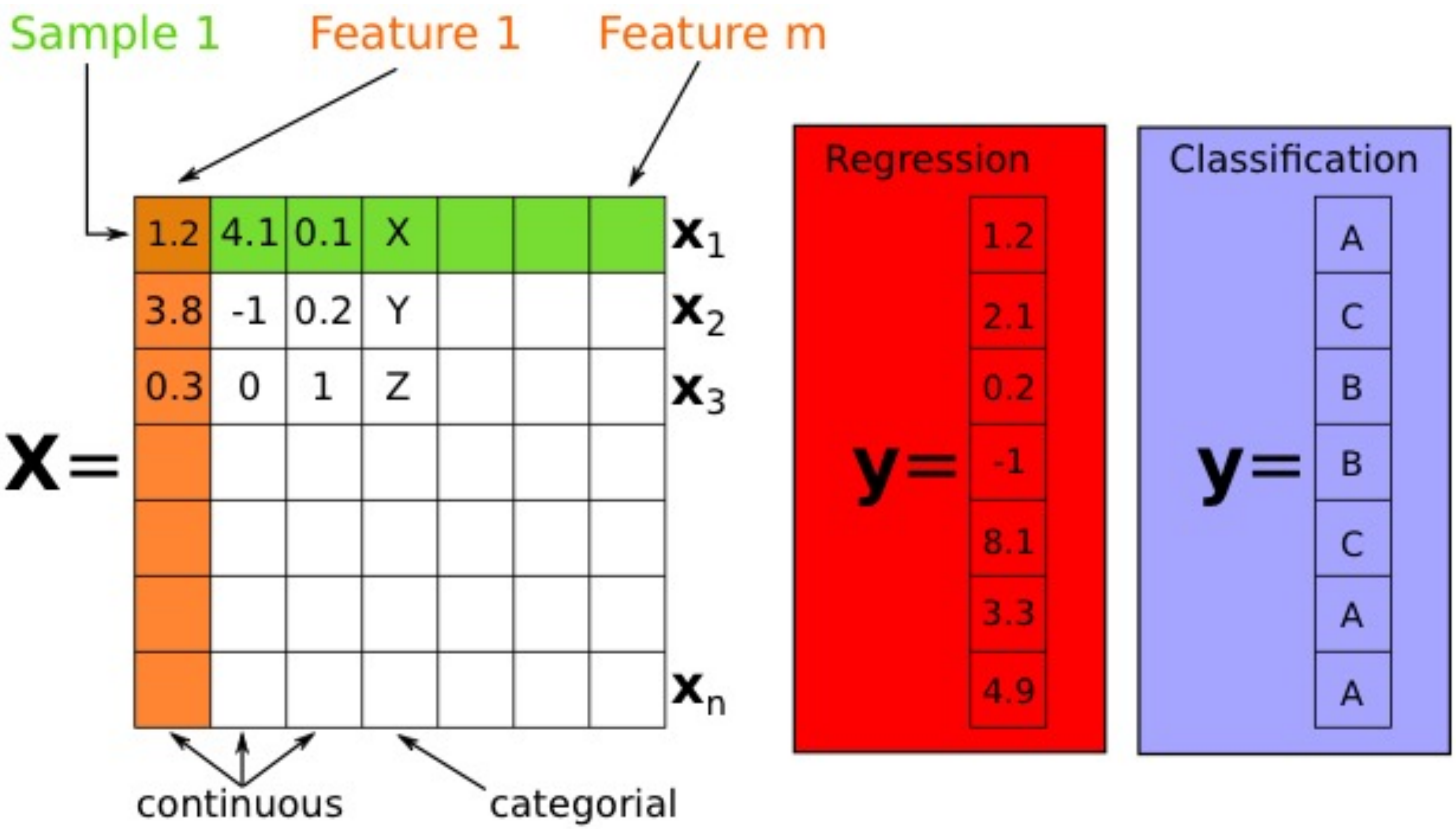
The Institute of Cancer Research
in partnership with
CANCER RESEARCH UK
The ROYAL MARSDEN NHS Foundation Trust

Linear Models

02/05/2018

Sebastian Pölsterl, Priya Narayanan, Shan Raza, Igor Kozin

Training Data



- In machine learning, we usually have a matrix of measurements/features $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ and a corresponding list of values $\boldsymbol{y} \in \mathbb{R}^{n}$ we want to predict.

A Dataset

Definitions

- A training **sample** x_i consists of m **features** $(x_{i1}, \dots, x_{im})^T$ and is associated with **output** y_i .
- Each feature and the output can either be **continuous** (a number) or **discrete** (from a predefined set of values).
- If the output is continuous, we perform regression and if it is discrete, classification.
- The **training set** $\mathcal{T} = (x_i , y_i)$ is comprised of *n* samples $(i = 1, \dots , n)$.
- Let $\boldsymbol{X}$ indicate a matrix where the *i*-th row corresponds to the *i*-th sample and $\boldsymbol{y} = (y_1, \dots, y_n)^T$ the vector of all outputs.

Body fat dataset

- Accurate measurement of body fat is inconvenient/costly and it is desirable to have easy methods of estimating body fat that are not inconvenient/costly.
- The dataset lists estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements for 252 men.
- If we are able to accurately predict body fat from easy to obtain circumference measurements, we found an easy way to estimate body fat that is cost-effective and convenient.

The Model

- Denote the *i*-th row of $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ as the *i*-th **feature vector** $\boldsymbol{x}_i \in \mathbb{R}^m$.
- By choosing a model, we determine how information captured by a feature vector $\boldsymbol{x}_i$ is used to form a prediction $\widehat{y}_i \in \mathbb{R}$.
- A machine learning **model** $\mathcal{F}(\boldsymbol{x}; \boldsymbol{\Theta})$ maps a feature vector $\boldsymbol{x}_i \in \mathbb{R}^m$ to a **prediction**

$$\widehat{y}_i = \mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\Theta}) \in \mathbb{R}$$

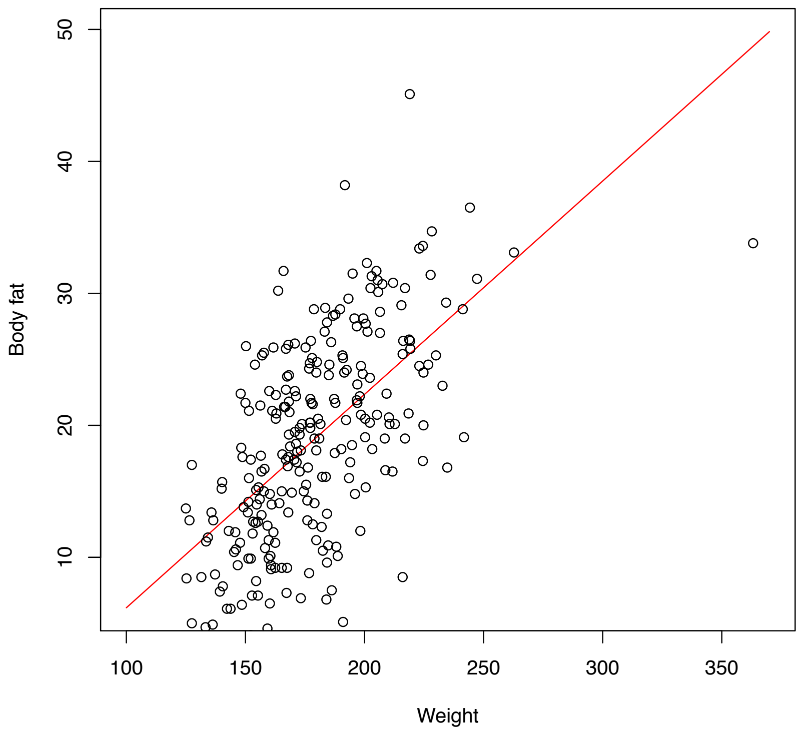
- $\boldsymbol{\Theta}$ is a set of **unknown parameters** we want to learn from data.

Linear Model
Definitions

Definition

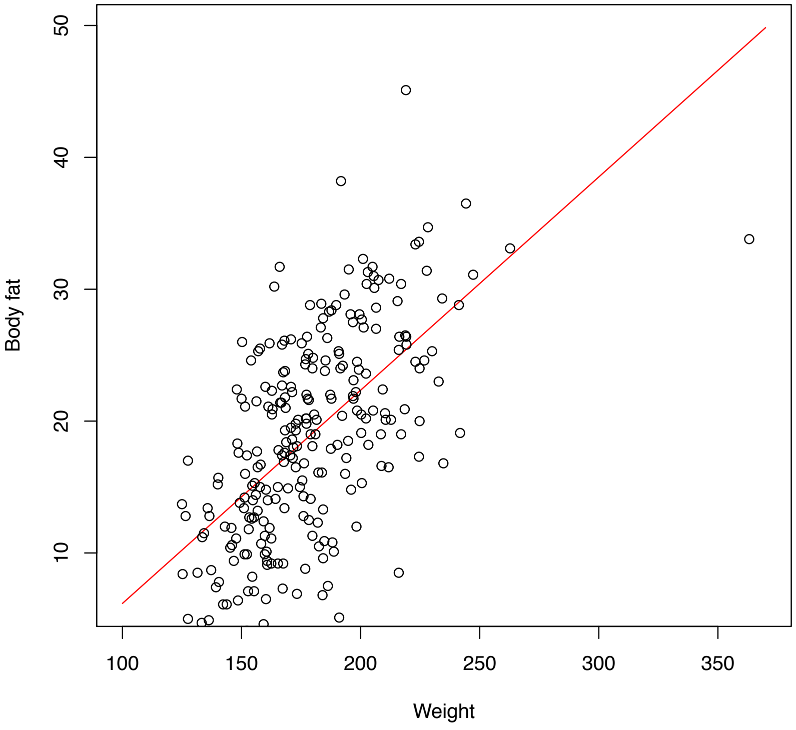
$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_m x_{i,m} + \epsilon_i = \beta_o + \boldsymbol{x}_i^\top \boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- The $\boldsymbol{\beta}$ parameters are **coefficients** or weights of the features.
- $\boldsymbol{\beta}$ is to be **estimated** from training data.
- The errors ϵ_i are independently and identically distributed (i.i.d.) with $E(\epsilon_i) = 0$ and $\mathrm{Var}(\epsilon_i) = \sigma^2$.



Linear Model
Coefficients

- Each **feature** is associated with one **coefficient** β_j.
- In addition, the coefficient β_0 denotes the **intercept** (or bias).
- Estimates are denoted by a **hat**: $\hat{\beta}_j$ denotes the estimate of the coefficient of the *j*-th feature.
- In the example to the right, $\beta_0 = -9.995$ (*y*-intercept) and $\beta_1 = 0.1617$ (slope; coefficient of *weight* feature).



The Loss Function

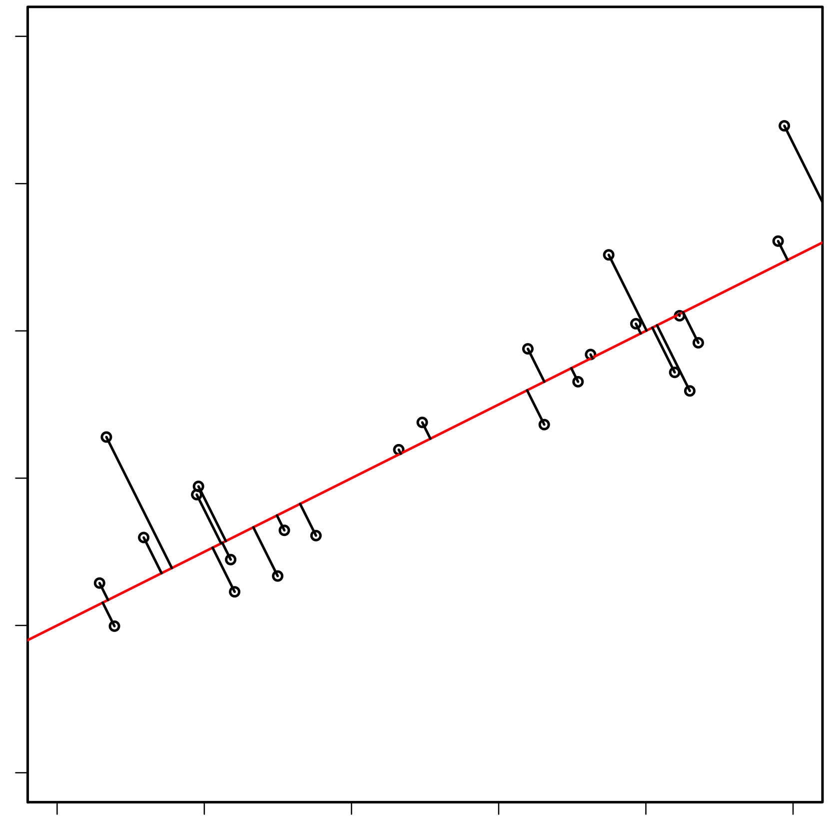
- **Training data** consists of a matrix of features $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ and a corresponding list of values $\boldsymbol{y} \in \mathbb{R}^{n}$ we want to predict.
- A **model** $\mathcal{F}(\boldsymbol{x}; \boldsymbol{\Theta})$ maps a feature vector $\boldsymbol{x}_i \in \mathbb{R}^{m}$ to a prediction $\widehat{y}_i = \mathcal{F}(\boldsymbol{x}_i; \boldsymbol{\Theta}) \in \mathbb{R}$ ($\boldsymbol{\Theta}$ are unknown parameters).
- We need to choose a function $\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ that measures how well our model is approximating our training data.

Linear Model
Loss Function

Definition (Linear Model)

$$\hat{y}_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_m x_{i,m} = \beta_o + \boldsymbol{x}_i^\top \boldsymbol{\beta}$$

- We need a way to asses how good our estimate $\hat{y}_i$ approximates the expected output y_i given the current estimates of the coefficients $\hat{\beta}_0, \dots, \hat{\beta}_m$.
- Hence, we define a **loss function** $\mathcal{L}(y, \hat{y})$.



Linear Model
Loss Function

Definition (Linear Model)

$$\widehat{y_i} = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \cdots + \beta_m x_{i,m} = \beta_o + \boldsymbol{x}_i^\top \boldsymbol{\beta}$$

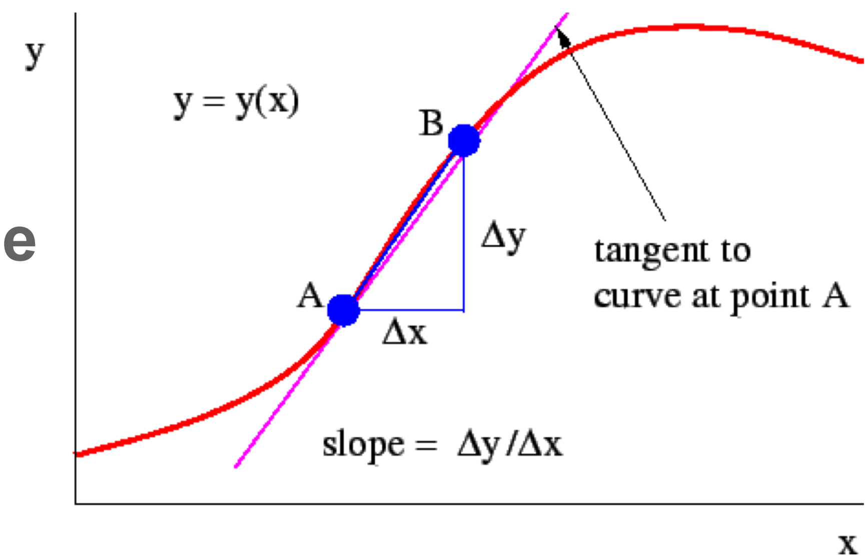
- If y_i is continuous such as ”amount of body fat”, a natural choice for the loss function is the **squared error**

$$\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 = \frac{1}{2}(\boldsymbol{y} - \widehat{\boldsymbol{y}})^\top(\boldsymbol{y} - \widehat{\boldsymbol{y}})$$

- $\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ gives the total loss over the whole training set.
- We want to choose the coefficients $\beta_0, \dots, \beta_m$ such that the total loss is **minimised**.
- Also referred to **residual sum of squares** in statistics.

Derivative

- Consider a function $y = f(x)$ with $x, y \in \mathbb{R}$.
- The derivative is denoted as $f'(x)$ or $\frac{d}{dx} f(x)$.
- The **derivative** gives the **slope** of $f(x)$ at the point x.
- It measures the sensitivity of $f(x)$ to small changes in the input x.



Partial Derivative and Gradient

- If the function has multiple inputs, we consider **partial derivatives**.
- The partial derivative $\frac{\partial}{\partial x_i} f(\boldsymbol{x})$ measures how sensitive $f(\boldsymbol{x})$ is to small changes in x_i alone at point $\boldsymbol{x}$.
- The **gradient** generalises the notion of derivative to the case where the derivative is with respect to a vector.
- The gradient of $f(\boldsymbol{x})$ is the vector containing all the partial derivatives, denoted $\nabla_{\boldsymbol{x}} f(\boldsymbol{x})$.

Function Minimisation

- We want to find parameters $\beta_0, \boldsymbol{\beta}$ such that $\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ is minimized, i.e. solving

$$\underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \;\; \mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$$

$$\underset{\beta_0, \boldsymbol{\beta}}{\arg\min} \;\; \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_o - \boldsymbol{x}_i^\top \boldsymbol{\beta} \;)^2$$

- $\mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}})$ reaches its minimum at the point β_j if its **partial derivative** with respect to β_j is zero for all $j = 0, \ldots, m$:

$$\frac{\partial}{\partial \beta_j} \mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = 0$$

Ordinary Least Squares
Estimation I

- Set the partial derivative to zero
$$\frac{\partial}{\partial \beta_j} \mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = -\sum_{i=1}^{n} x_{ij} \left(y_i - \beta_0 - \boldsymbol{x}_i^\top \boldsymbol{\beta} \right) = 0$$
- Gradient vector in matrix notation:
$$\nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = -\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \boldsymbol{0}$$
- **Note**: $\beta = (\beta_0, \dots, \beta_m)^\top$ and the first column of $\boldsymbol{X}$ contains only 1 to accommodate the intercept β_0, i.e. $\boldsymbol{X}$ is a $n \times m + 1$ matrix.

Ordinary Least Squares
Estimation II

Definition (Ordinary Least Squares Estimate)

$$(\boldsymbol{X}^\top \boldsymbol{X})\boldsymbol{\beta} = \boldsymbol{X}^\top \boldsymbol{y}$$

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

- The minimum of the loss function is unique.
- Estimates of the coefficients can be obtained in closed form and therefore no iterative optimisation is required.
- $\boldsymbol{X}$ must have full column rank, i.e., $\boldsymbol{X}^\top \boldsymbol{X}$ is positive definite.
- Prediction is performed by

$$\hat{y}_\mathrm{i} = \hat{\beta}_0 + \hat{\beta}_1 x_{i,1} + \hat{\beta}_2 x_{i,2} + \cdots + \hat{\beta}_m x_{i,m}$$

Exercise I

- Open the notebook **01_linear_regression.ipynb**

in partnership with

Logistic Regression

02/05/2018

Sebastian Pölsterl, Priya Narayanan, Shan Raza, Igor Kozin

Logistic Regression

- Consider a binary classification problem where $y_i \in \{0, 1\}$.
- If $y_i = 1$, the *i*-th sample belongs to the **positive class**, otherwise to the **negative class**.
- Create a model of the probability of samples $\boldsymbol{x}_i$ belonging to the positive class

$$\pi_i = P(y_i = 1 \mid x_{i,1}, \dots, x_{i,m})$$

- Remember that the linear model is define as

$$\eta_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_m x_{i,m}$$

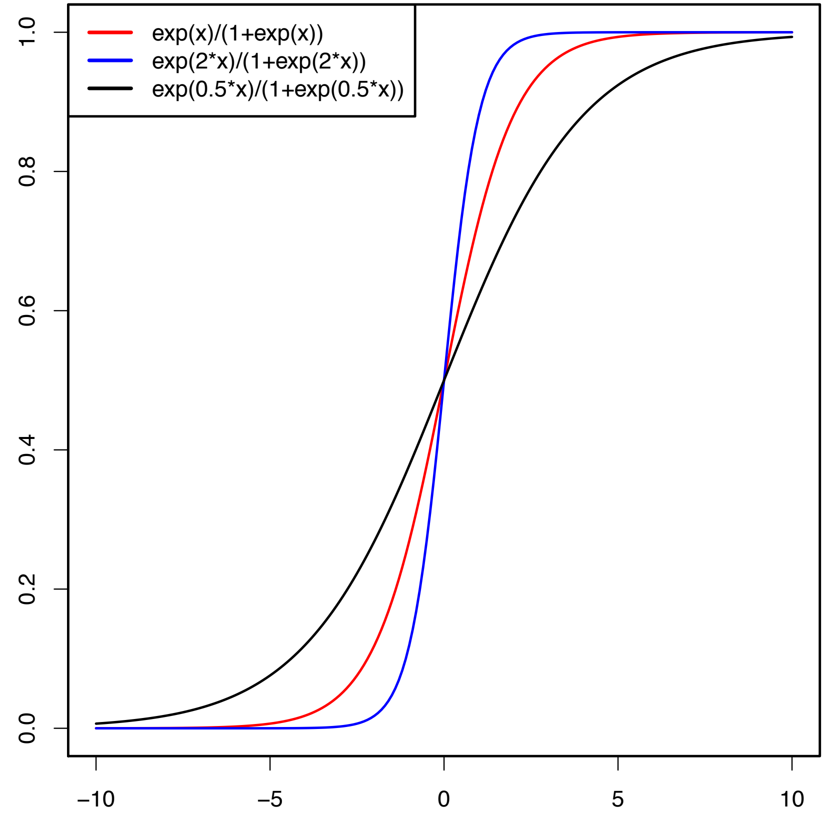
- **How to connect the probability π_i to the linear predictor η_i?**

Logistic Regression

Response function

- The **logistic function** $h(x)$ connects the probability π_i to the linear predictor η_i

$$\pi_i = h(\eta_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$$



Logistic Regression
Loss function

Definition (Likelihood function)

$$\mathcal{L}(\beta_o, \boldsymbol{\beta}; \boldsymbol{X}) = \prod_{i=1}^{n} P(\, y_i \mid \boldsymbol{x}_i \,) = \prod_{i=1}^{n} \pi_i^{y_i} (1 - \pi_i)^{1-y_i}$$

- We want **maximise** the probabilities across the whole training data.

Definition (Maximum Likelihood Estimate; MLE)

$$\widehat{\boldsymbol{\beta}} = \mathrm{argmax}_{\beta_0, \boldsymbol{\beta}} \log \mathcal{L}(\beta_o, \boldsymbol{\beta}; \boldsymbol{X})$$

Logistic Regression
Iterative Optimisation

- We proceed as before, by setting the gradient of $\mathcal{L}(\beta_o, \boldsymbol{\beta}; \boldsymbol{X})$ to zero

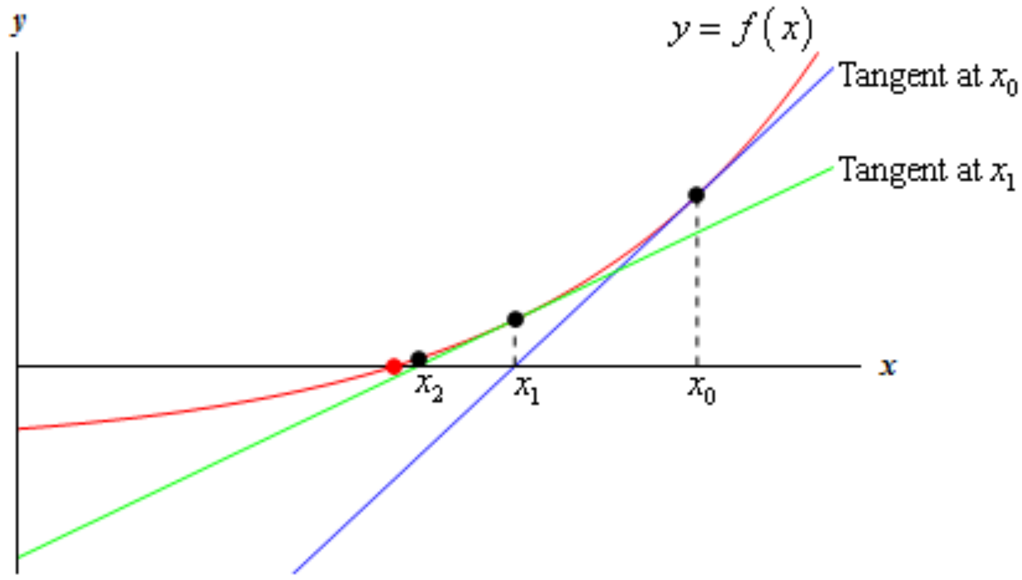
$$\nabla_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta}) = \boldsymbol{X}^{\top}(\boldsymbol{y} - \boldsymbol{\pi}) = \mathbf{0}$$

- However, we note that there is no closed form solution to finding parameters maximising the likelihood function.
- We need to **iteratively** search for the optimal set of parameters.

Newton's method

- Given a function $f(x)$ and its derivative $f'(x)$, **Newton's method** aims to find the value x that satisfies $f(x) = 0$ by iteratively approximating the solution by calculating

$$x^{(t+1)} = x^{(t)} - \frac{f(x^{(t)})}{f'(x^{(t)})}$$



Logistic Regression
Iterative Optimisation

- We want to find β_o and $\boldsymbol{\beta}$ for which

$$\nabla_{\boldsymbol{\beta}}\mathcal{L}(\boldsymbol{\beta}) = \boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{\pi}) = \mathbf{0}$$

- Using Newton's method, we find that the update step has the form

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left(\nabla^2_{\boldsymbol{\beta}}\mathcal{L}\left(\boldsymbol{\beta}^{(t)}\right)\right)^{-1}\nabla_{\boldsymbol{\beta}}\mathcal{L}\left(\boldsymbol{\beta}^{(t)}\right)$$

- Starting from an initial guess $\boldsymbol{\beta}^{(0)}$, we iteratively update our estimate of $\boldsymbol{\beta}$ until convergence.
- Usually, the starting point is the zero vector, $\boldsymbol{\beta}^{(0)} = \mathbf{0}$.

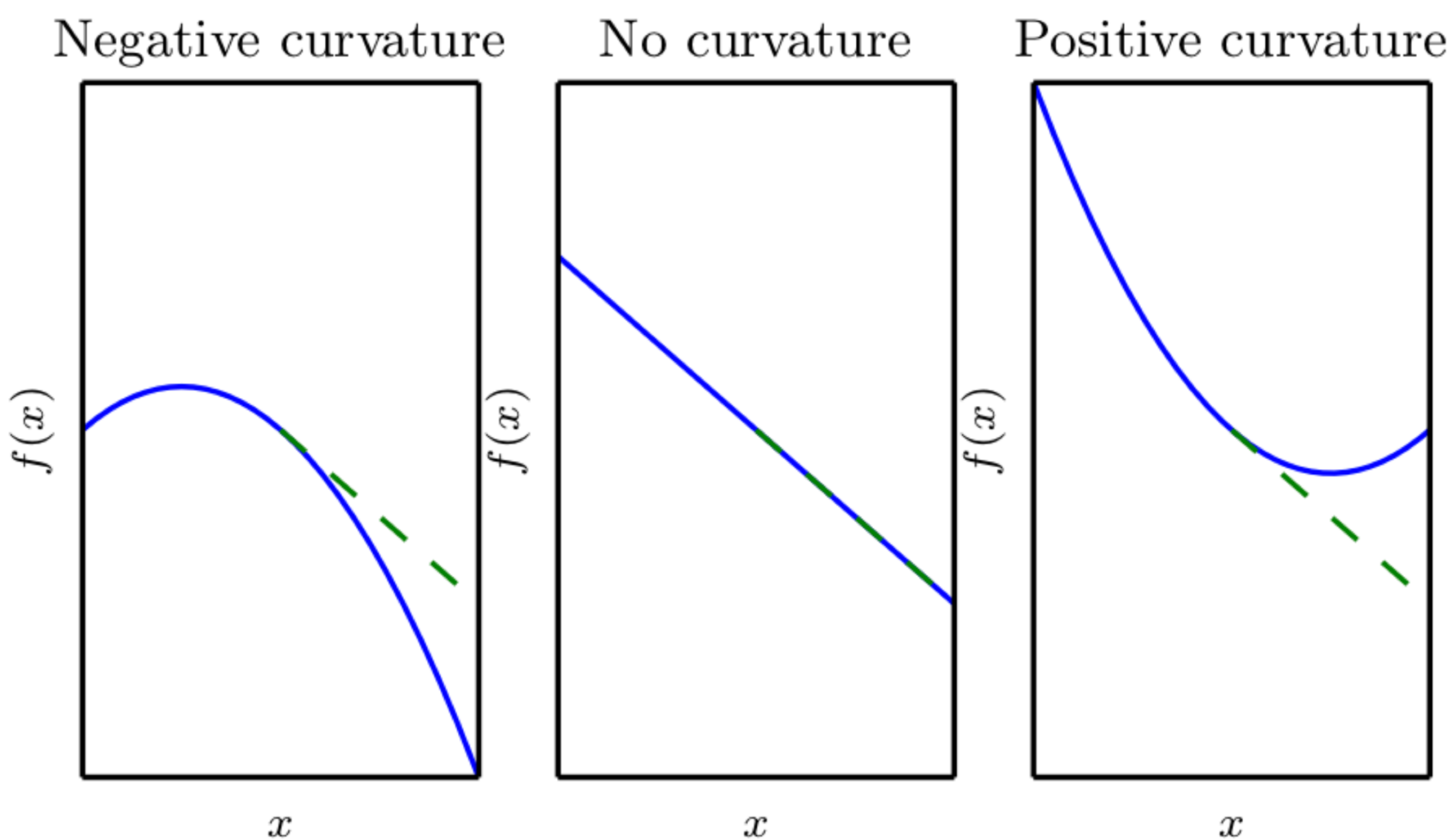
Second Derivative
Hessian

- The derivative of a derivative is called a **second-order derivative**.
- The **Hessian** matrix $\mathbf{H} = \nabla_{\boldsymbol{\beta}}^2 \mathcal{L}\left(\boldsymbol{\beta}^{(t)}\right)$ denotes a matrix of partial second-order derivatives

$$(\mathbf{H})_{\mathrm{i,j}} = \frac{\partial^2}{\partial \beta_i \partial \beta_j} \mathcal{L}\left(\boldsymbol{\beta}^{(t)}\right)$$

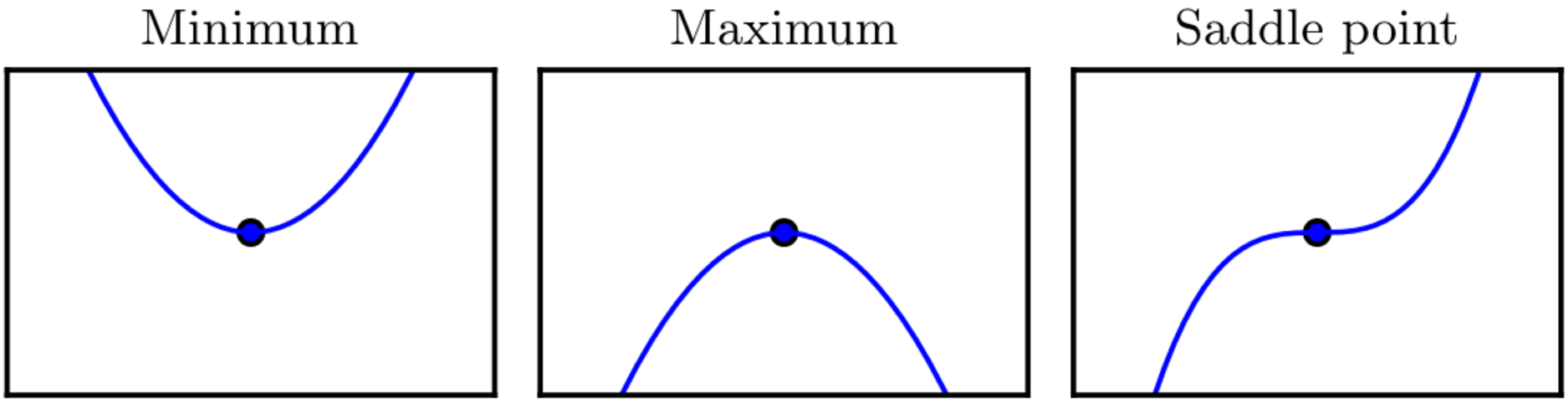
Second Derivative
Curvature

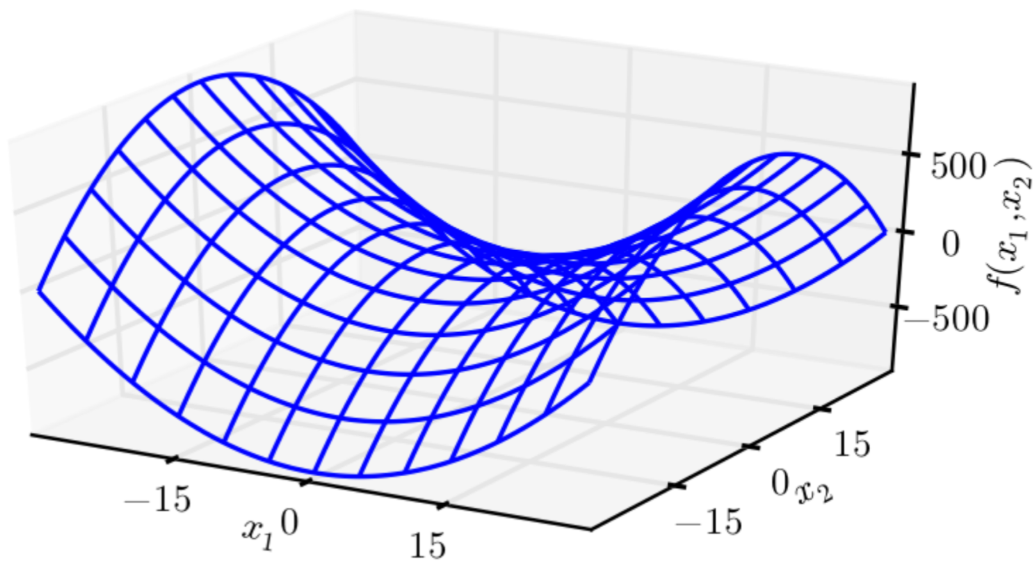
- The second-order derivative measures the **curvature** at a given point.



Newton's Method

Critical Points





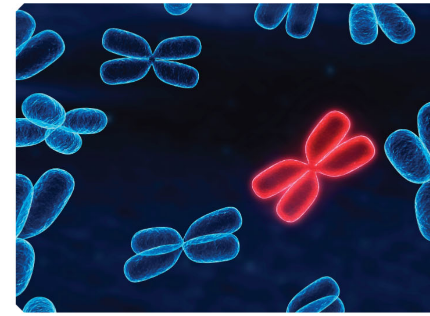
Exercise II

- Open the notebook **02_logistic_regression.ipynb**

ICR

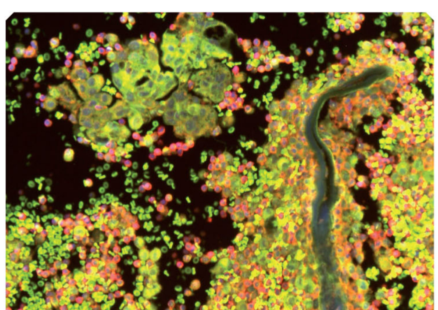
Unrivalled track record

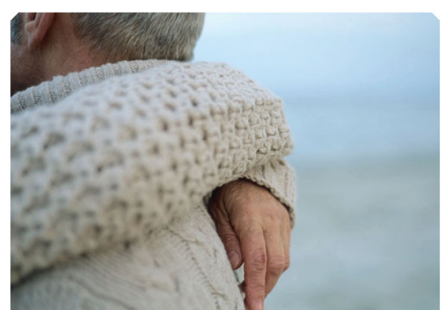
ICR The Institute of Cancer Research



Making the discoveries that defeat cancer

ICR





One of the world's most influential cancer research institutes