Linear and Logistic Regression

- Supervised learning methods

- Linear regression: predict a **continuous** target y
- Logistic regression: predict a **categorical** target y (label, class value)
  - **Classification** and not regression
  - Classification = recognition
  - *Binary* classification, to be precise (e.g., Yes/No, -1/1, 0/1)
  - Extensions to multi-class later in the course

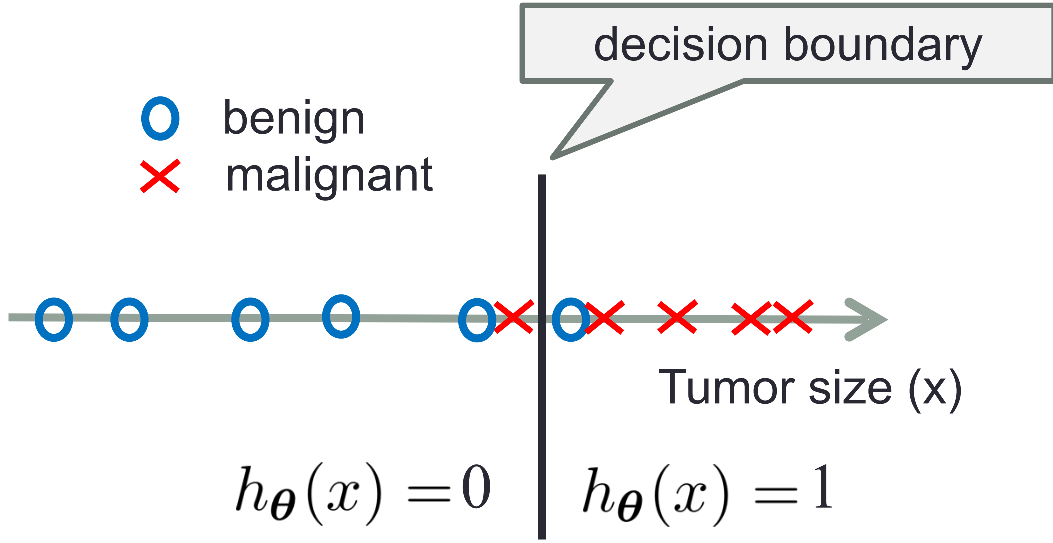
  - Interpretability

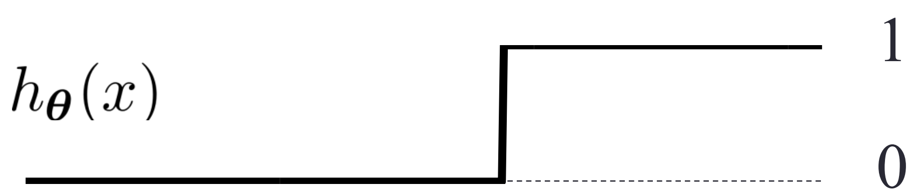
Example (step function hypothesis)

labeled data

i	Tumor size (mm) x	Malignant? y
1	2.3	0 (N)
2	5.1	1 (Y)
3	1.4	0 (N)
4	6.3	1 (Y)
5	5.3	1 (Y)
	…	…

labels



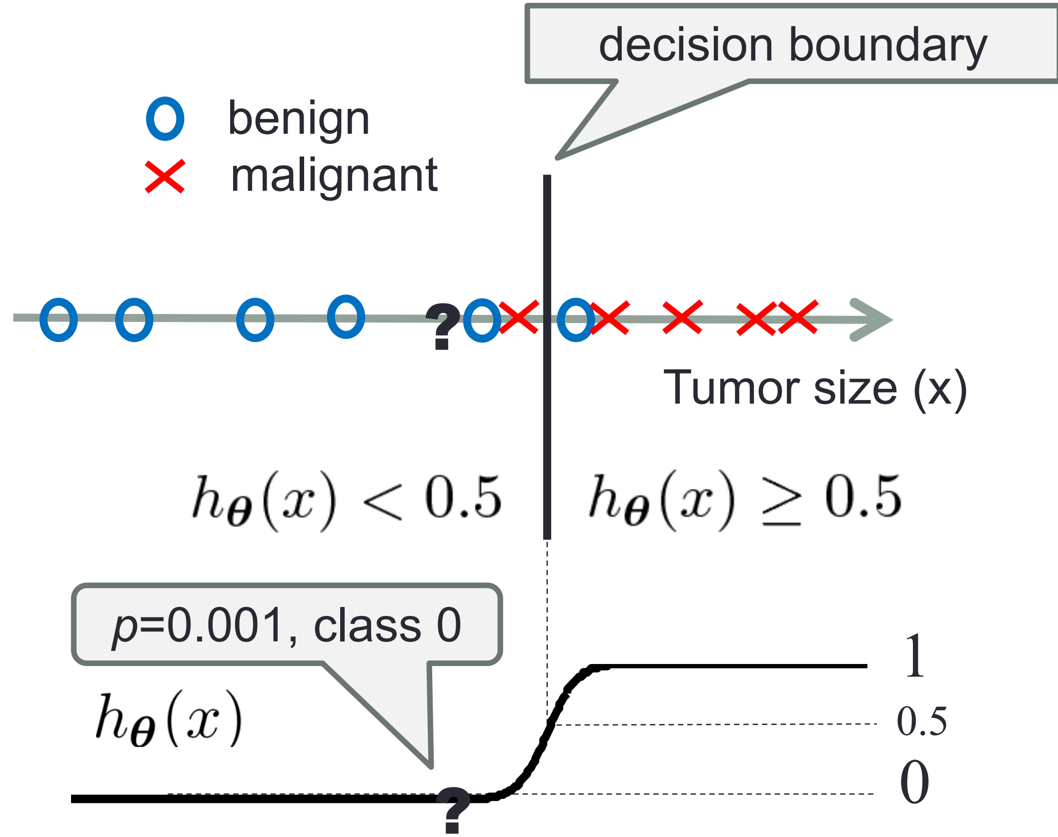


Example (logistic function hypothesis)

labeled data

i	Tumor size (mm) x	Malignant? y
1	2.3	0 (N)
2	5.1	1 (Y)
3	1.4	0 (N)
4	6.3	1 (Y)
5	5.3	1 (Y)
	…	…

labels



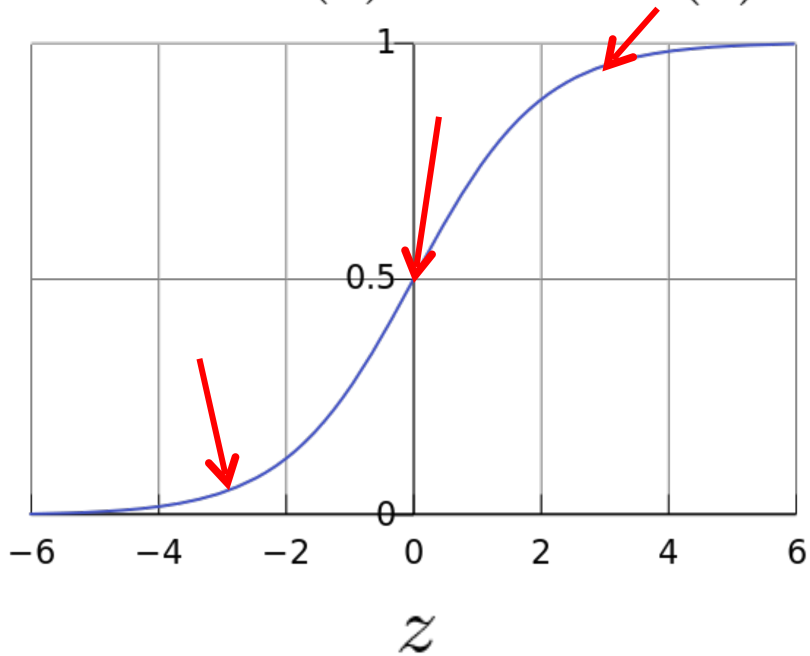
Hypothesis: Tumor is malignant with **probability p**

Classification: if $p < 0.5$: 0
if $p \geq 0.5$: 1

Logistic (sigmoid) function

$$\sigma(-3) \approx 0.05 \quad \sigma(0) = 0.5 \quad \sigma(3) \approx 0.95$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



- Advantages over step function for classification:
  - Differentiable → Gradient Descent can be used
  - Contains additional information (how certain the prediction is)

Logistic regression hypothesis

1. Reduce high-dimensional input $\boldsymbol{x}$ to a scalar

$$
\begin{aligned}
z &= \boldsymbol{x}^T \boldsymbol{\theta} \\
&= \theta_0 + \theta_1 \cdot x_1 + \cdots + \theta_n \cdot x_n
\end{aligned}
$$

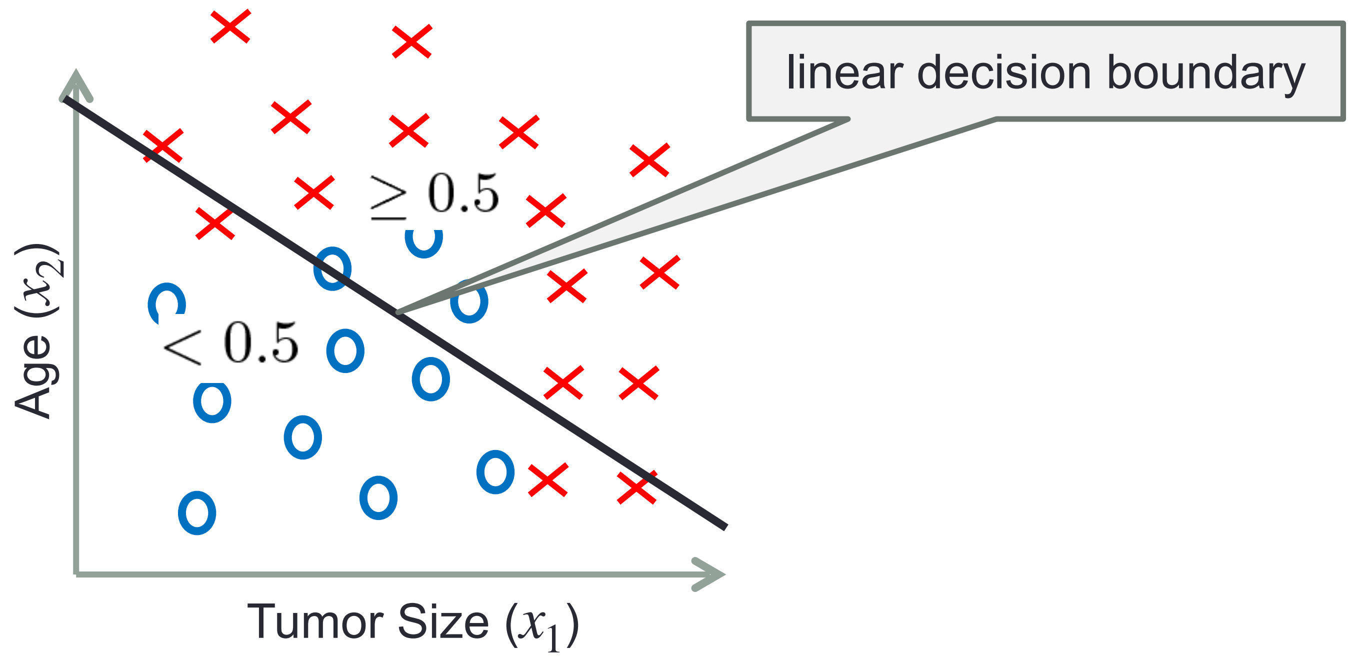
2. Apply logistic function

$$
\begin{aligned}
h_{\boldsymbol{\theta}}(\boldsymbol{x}) &= \sigma(\boldsymbol{x}^T \boldsymbol{\theta}) \\
&= \sigma(\theta_0 + \theta_1 \cdot x_1 + \cdots + \theta_n \cdot x_n)
\end{aligned}
$$

3. Interpret output $h_{\boldsymbol{\theta}}(\boldsymbol{x})$ as probability and predict class:

$$
\textit{Class} = \begin{cases} 0 & \text{if } h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5 \\ 1 & \text{if } h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5 \end{cases}
$$

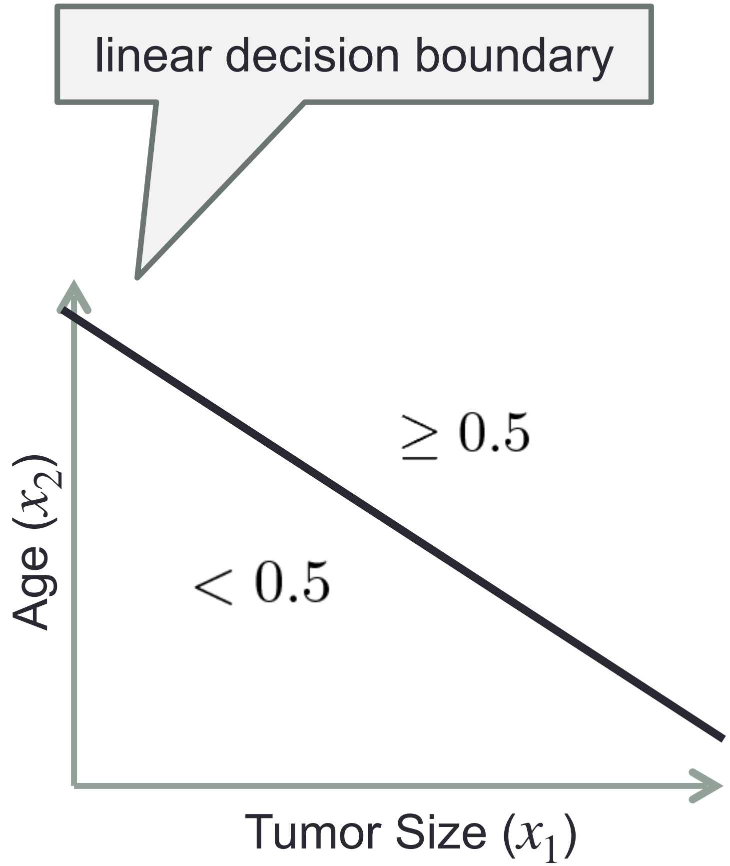
Linear features

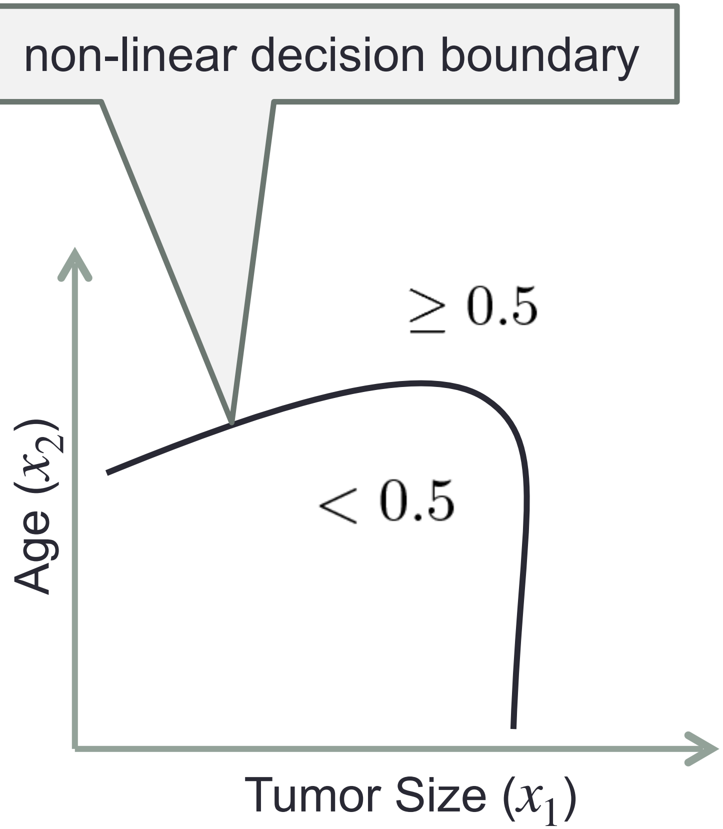


$$x_1 = \text{Tumor Size},\ x_2 = \text{Age}$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(-10 + 2 \cdot x_1 + 0.05 \cdot x_2)$$

Decision boundaries





$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(-10 + 2 \cdot x_1 + 0.05 \cdot x_2)$$

$$h_{\boldsymbol{\theta}}(\boldsymbol{\phi}) = \sigma(-3 + 1.2 \cdot \phi_1 + 0.07 \cdot \phi_2 - 0.9 \cdot \phi_3 + \ldots)$$

Decision boundary is a property of hypothesis, not of data!
(because the decision boundary determines how the classes will be split)

Non-linear features

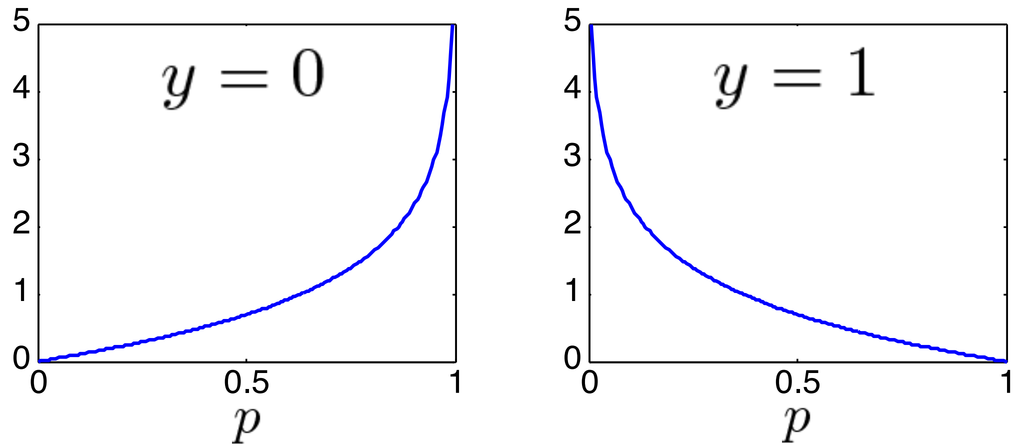
- Similarly as for linear regression, we use the design matrix X
- Non-linear features: Any non-linear transformation of x that is included in the design matrix

- **Binary features** are also possible, e.g.,
  - Age > 18,
  - $x_1 > 5$ and $x_1 < 10$
  - $x_2 == 25$

Logistic regression cost function

- How well does the hypothesis $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{x}^T\boldsymbol{\theta})$ fit the data?
- „Cost“ for predicting probability *p* ($p = h_\theta(x)$) when the real value (target, label) is *y*:

$$\mathrm{Cost}(p,\ y) = \begin{cases} -\log(1-p) & \text{if } y = 0 \ , \\ -\log(p) & \text{if } y = 1 \ . \end{cases}$$



- Mean over all training examples: $J(\boldsymbol{\theta}) = \dfrac{1}{m}\displaystyle\sum_{i=1}^{m} \mathrm{Cost}(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}),\ y^{(i)})$

Minimizing the cost via gradient descent

- Gradient of logistic regression cost:

$$\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \Big(\underbrace{h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}}_{\text{„error“}} \Big) \cdot \underbrace{x_j^{(i)}}_{\text{„input“}}$$

(for j=0: $x_0^{(i)} = 1$)

Gradient descent algorithm

- Repeat until convergence

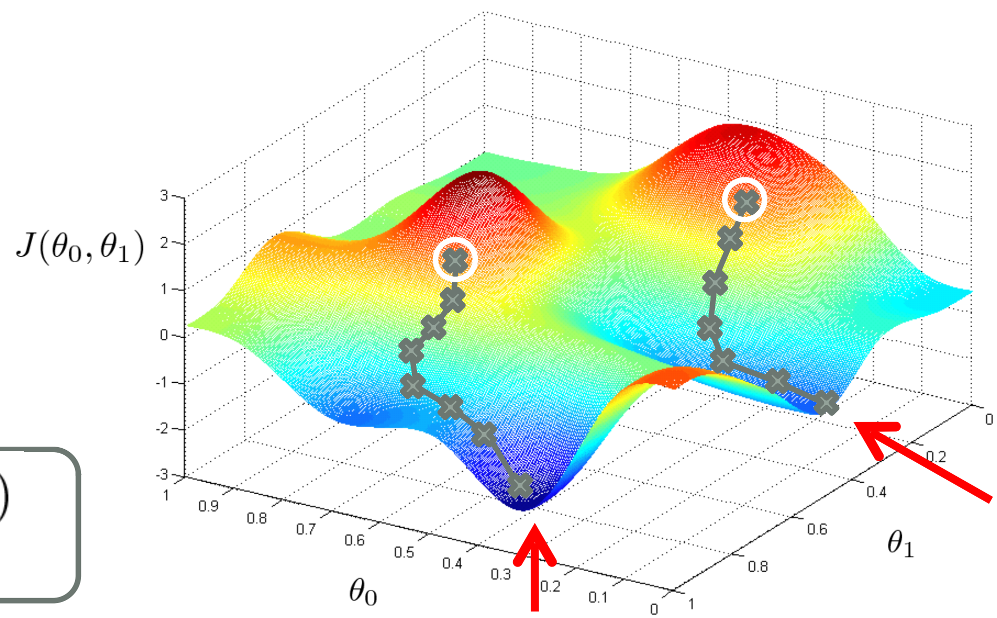
$$\theta_j := \theta_j - \eta \cdot \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(***simultaneously*** *updating* θ_0 *and* θ_1 *)*

negative gradient = descent

learning rate („eta“)

partial derivative of $J(\theta_0, \theta_1)$ with respect to θ_j



Linear vs. Logistic Regression

Linear Regression

- Regression
- Hypothesis $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\theta}$
- Cost for one training example:

$$\mathrm{Cost}(h,\ y) = (h-y)^2$$

- Gradient

$$\frac{\partial}{\partial\theta_j}J(\boldsymbol{\theta}) = \frac{2}{m}\sum_{i=1}^{m}\underbrace{\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}\right)}_{\text{„error“}} \cdot \underbrace{x_j^{(i)}}_{\text{„input“}}$$

- Analytical:

$$\boldsymbol{\theta}^* = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Logistic Regression

- Binary classification (!)
- Hypothesis $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{x}^T\boldsymbol{\theta})$
- Cost for one training example:

$$\mathrm{Cost}(p,\ y) = \begin{cases} -\log(1-p) & \text{if } y=0\ , \\ -\log(p) & \text{if } y=1\ . \end{cases}$$

- Gradient

$$\frac{\partial}{\partial\theta_j}J(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\underbrace{\left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}\right)}_{\text{„error“}} \cdot \underbrace{x_j^{(i)}}_{\text{„input“}}$$

- No analytical solution!