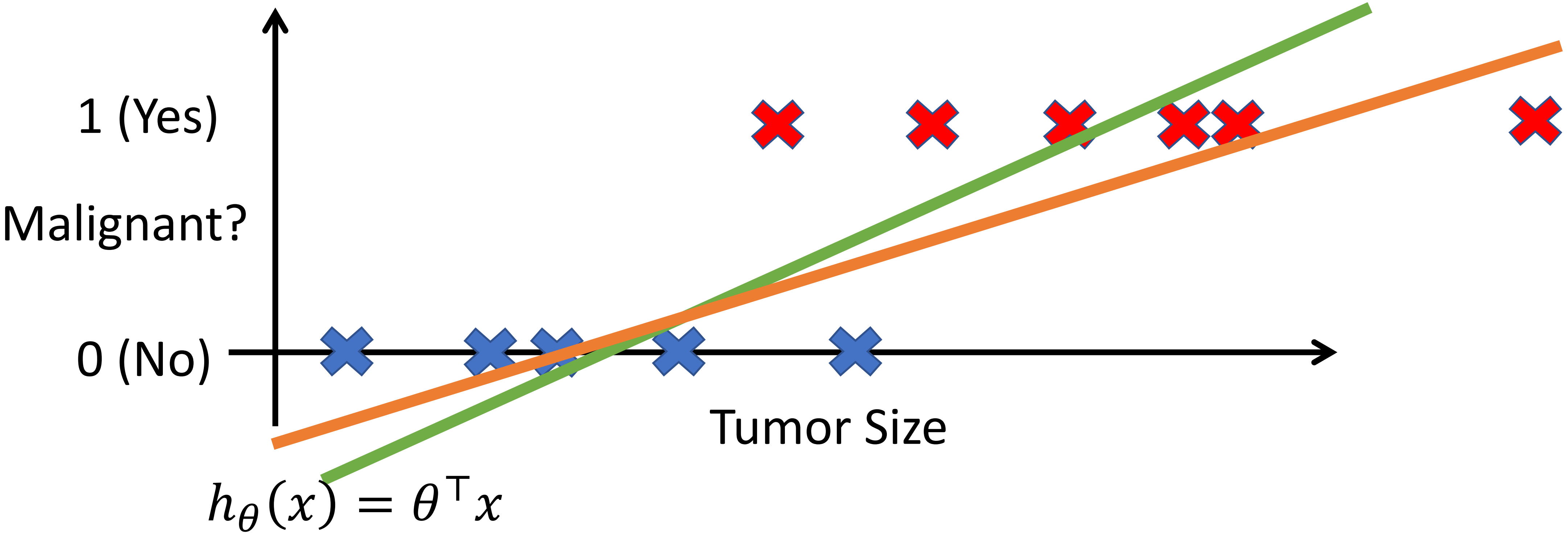
Logistic Regression

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- Regularization
- Multi-class classification

Logistic Regression

- **Hypothesis representation**
- Cost function
- Logistic regression with gradient descent
- Regularization
- Multi-class classification



- Threshold classifier output $h_\theta(x)$ at 0.5
  - If $h_\theta(x) \geq 0.5$, predict "$y = 1$"
  - If $h_\theta(x) < 0.5$, predict "$y = 0$"

Classification: $y = 1$ or $y = 0$

$h_\theta(x) = \theta^\top x$ (from linear regression)
can be > 1 or < 0

Logistic regression: $0 \leq h_\theta(x) \leq 1$

Logistic regression is actually for **classification**

Hypothesis representation

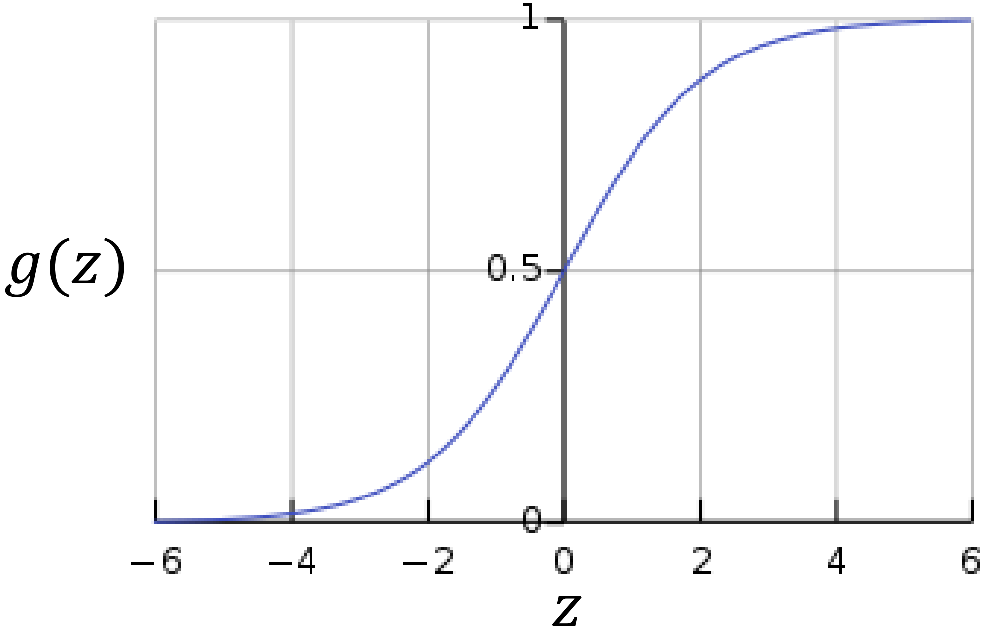
- Want $0 \leq h_\theta(x) \leq 1$

- $h_\theta(x) = g(\theta^\top x)$,

  where $g(z) = \frac{1}{1+e^{-z}}$

- Sigmoid function
- Logistic function

$$\boxed{h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}}$$



Interpretation of hypothesis output

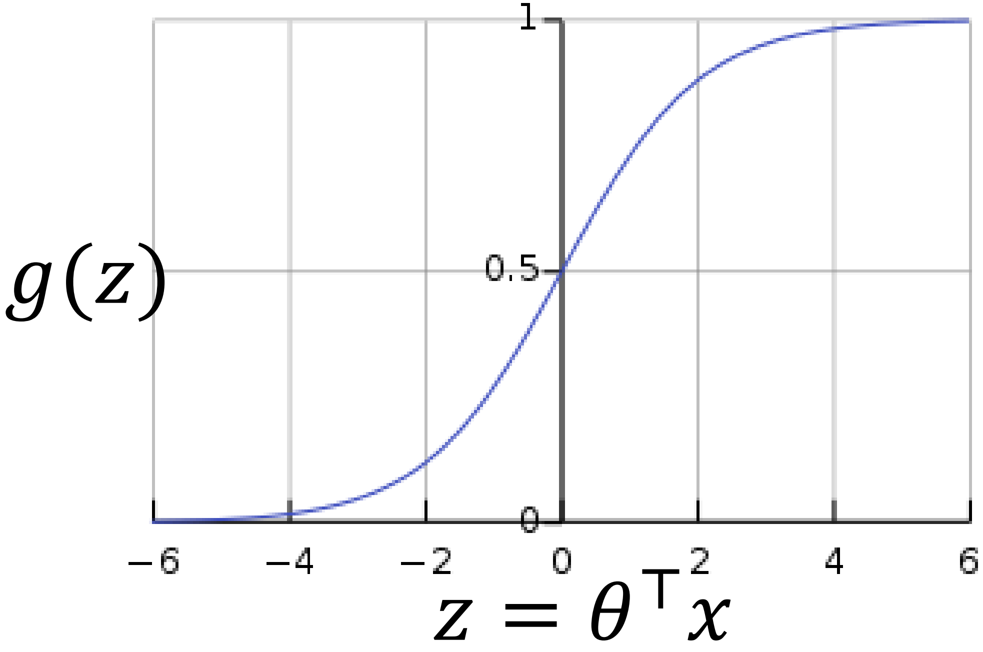
- $h_\theta(x)$ = estimated probability that $y = 1$ on input x

- Example: If $x = \begin{bmatrix} x_0 \\ \mathrm{x}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \mathrm{tumorSize} \end{bmatrix}$
- $h_\theta(x) = 0.7$

- Tell patient that 70% chance of tumor being malignant

Logistic regression

$$h_\theta(x) = g(\theta^\top x)$$
$$g(z) = \frac{1}{1 + e^{-z}}$$



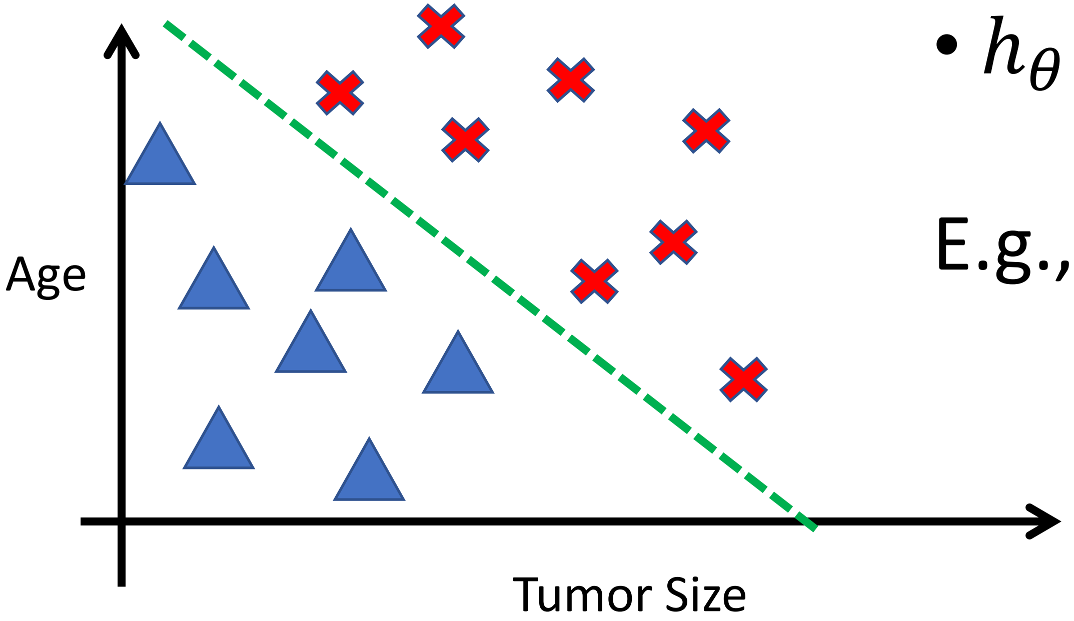
Suppose predict "y = 1" if $h_\theta(x) \geq 0.5$

$$z = \theta^\top x \geq 0$$

predict "y = 0" if $h_\theta(x) < 0.5$

$$z = \theta^\top x < 0$$

Decision boundary



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

E.g., $\theta_0 = -3, \theta_1 = 1, \ \theta_2 = 1$

- Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

Logistic Regression

- Hypothesis representation
- **Cost function**
- Logistic regression with gradient descent
- Regularization
- Multi-class classification

Training set with m examples

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$$

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad x_0 = 1, y \in \{0, 1\}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}$$

How to choose parameters θ?

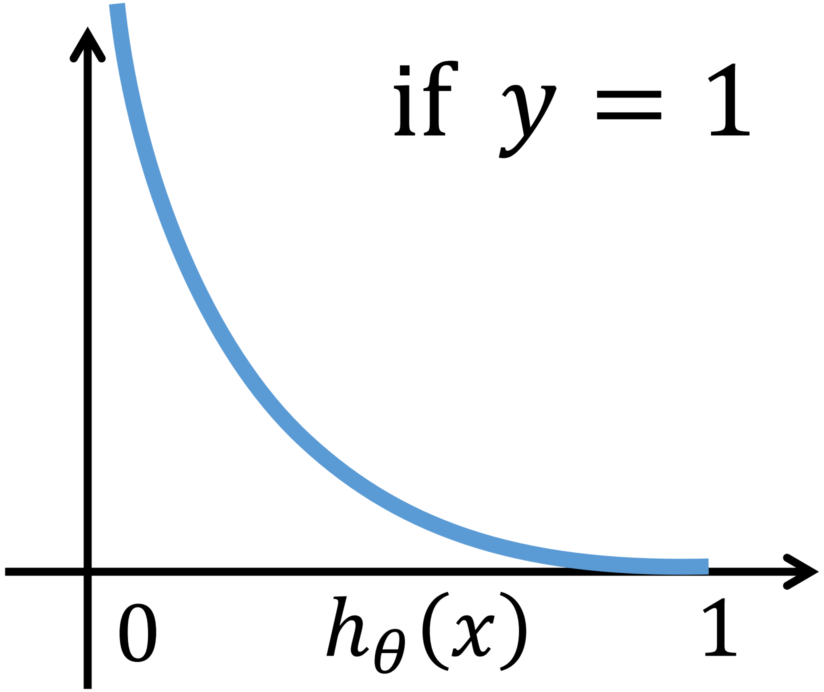
Cost function for **Linear Regression**

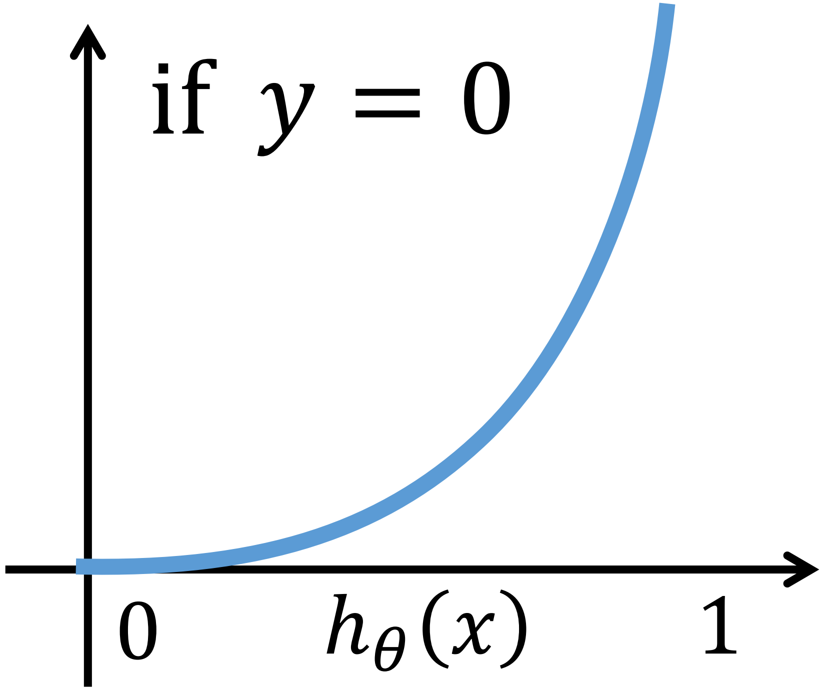
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(h_\theta(x^{(i)}), y))$$

$$\boxed{\mathrm{Cost}(h_\theta(x), y) = \frac{1}{2}(h_\theta(x) - y)^2}$$

Cost function for **Logistic Regression**

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log\big(h_\theta(x)\big) & \text{if } y = 1 \\ -\log\big(1 - h_\theta(x)\big) & \text{if } y = 0 \end{cases}$$





Logistic regression cost function

- $\text{Cost}(h_\theta(x), y) = \begin{cases} -\log\big(h_\theta(x)\big) & \text{if } y = 1 \\ -\log\big(1 - h_\theta(x)\big) & \text{if } y = 0 \end{cases}$

$$\Downarrow$$

- $\text{Cost}(h_\theta(x), y) = -y \ \log\big(\mathrm{h}_\theta(\mathrm{x})\big) - (1 - \mathrm{y}) \log\big(1 - h_\theta(x)\big)$

- If $y = 1$: $\text{Cost}(h_\theta(x), y) = -\log\big(h_\theta(x)\big)$
- If $y = 0$: $\text{Cost}(h_\theta(x), y) = -\log\big(1 - h_\theta(x)\big)$

Logistic regression

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)}))$$

$$= -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

Learning: fit parameter θ

$$\min_\theta J(\theta)$$

Prediction: given new x

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^\top x}}$

Logistic Regression

- Hypothesis representation
- Cost function
- **Logistic regression with gradient descent**
- Regularization
- Multi-class classification

Derivation of Gradient Descent for Binary Logistic Regression.

$$\frac{\partial J}{\partial \theta} = \frac{\partial J}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial \theta}$$

where $\hat{y} = \dfrac{1}{1+e^{-z}} = h(x)$.

$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

$J = -\left[y \log \hat{y} + (1-y) \log(1-\hat{y}) \right]$

→ So, Now let us start with

$$\frac{\partial J}{\partial \theta}$$

⇒ Consider this derivation $z = \theta_0 + \theta_1 x_1 + \dots \theta_n x_n = \theta^T x$

In vector form it can also be written as

$$z = \vec{\theta} \cdot \vec{x}.$$

Now, $\dfrac{\partial z}{\partial \theta_0} = 1(x_0)$ $\dfrac{\partial z}{\partial \theta_1} = x_1$ $\dfrac{\partial z}{\partial \theta_2} = x_2$ .. and so on

So, $\boxed{\dfrac{\partial z}{\partial \theta} = X}$ In Generalized form.

since $\boxed{z = \theta^T x}$

→ How $J = -\left[y \log \hat{y} + (1-y) \log(1-\hat{y})\right]$

So, $$\frac{\partial J}{\partial \hat{y}} = -\left[y \cdot \frac{\partial}{\partial \hat{y}} \log \hat{y} + (1-y) \frac{\partial}{\partial \hat{y}} \log(1-\hat{y}) \right]$$

$$\therefore \frac{\partial J}{\partial \hat{y}} = -\left[y \cdot \frac{1}{\hat{y}} + \frac{(1-y)}{(1-\hat{y})} \right]$$

And $$\frac{\partial \hat{y}}{\partial z} = \frac{\partial}{\partial z} \left(\frac{1}{1+e^{-z}} \right)$$

$$= \frac{\partial}{\partial z} (1+e^{-z})^{-1} = (-1)(1+e^{-z})^{-2} \cdot e^{-z}$$

By chain rule

$$\therefore \frac{\partial \hat{y}}{\partial z} = e^{-z}\left(1+e^{-z}\right)^{-2} = \frac{e^{-z}}{\left(1+e^{-z}\right)^{2}}$$

$$\frac{\partial \hat{y}}{\partial z} = \overbrace{\left(\frac{1}{1+e^{-z}}\right)}^{\hat{y}}\overbrace{\left(\frac{e^{-z}}{1+e^{-z}}\right)}^{(1-\hat{y})}$$

$$\therefore \frac{\partial \hat{y}}{\partial z} = \hat{y}\,(1-\hat{y})$$

So, $$\frac{\partial J}{\partial \theta} = -\left[\frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}\right] \hat{y}\,(1-\hat{y})\,(x)$$

$\Rightarrow$

$$\frac{\partial J}{\partial \theta} = -\left[\frac{y(1-\hat{y}) - (1-y)(\hat{y})}{(1-\hat{y})(\hat{y})}\right](\hat{y})(1-\hat{y}) \cdot x.$$

$$\frac{\partial J}{\partial \theta} = -\left[y - y\hat{y} - \hat{y} + y\hat{y}\right] x.$$

$$\frac{\partial J}{\partial \theta} = -(y-\hat{y})x.$$

$$\therefore \frac{\partial J}{\partial \theta} = (\hat{y}-y)x$$

↳ For single training sampe

Gradient descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

Goal: $\min_\theta J(\theta)$ Convex function!

Repeat {

$$\theta_j \coloneqq \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

(Simultaneously update all θ_j)

$$\boxed{\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta\left(x^{(i)}\right) - y^{(i)})\, x_j^{(i)}}$$

Gradient descent

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\ \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

Goal: $\min_{\theta} J(\theta)$

Repeat { (Simultaneously update all θ_j)

$$\theta_j \coloneqq \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

}

Gradient descent for Linear Regression

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} \qquad \boxed{h_\theta(x) = \theta^\top x}$$

}

Gradient descent for Logistic Regression

Repeat {

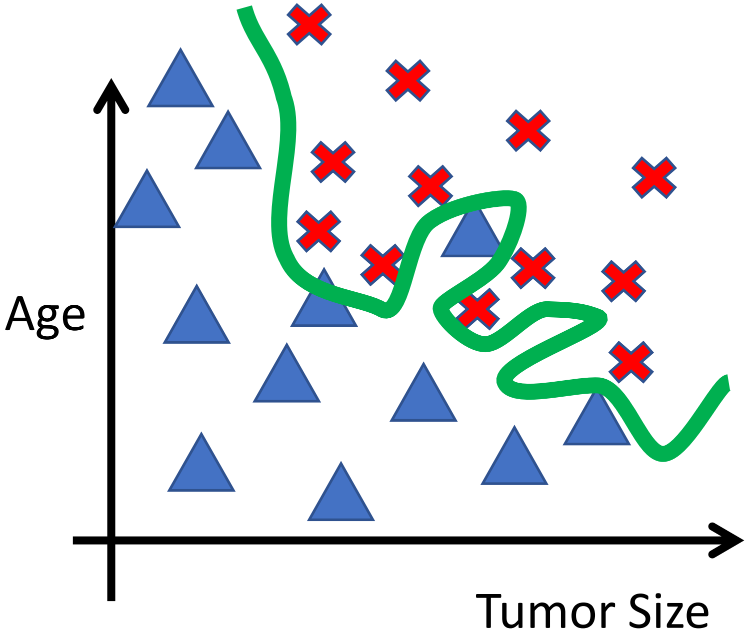
$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} \qquad \boxed{h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}}$$

}

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- **Regularization**
- Multi-class classification

Regularized logistic regression



$$h_\theta(x) = g(\theta_0 + \theta_1 x + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \theta_7 x_1 x_2^3 + \cdots)$$

- Cost function:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right) + \frac{\lambda}{2}\sum_{j=1}^{n} \theta_j^2\right]$$

Gradient descent (Regularized)

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^\top x}}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} + \lambda \theta_j\right]$$

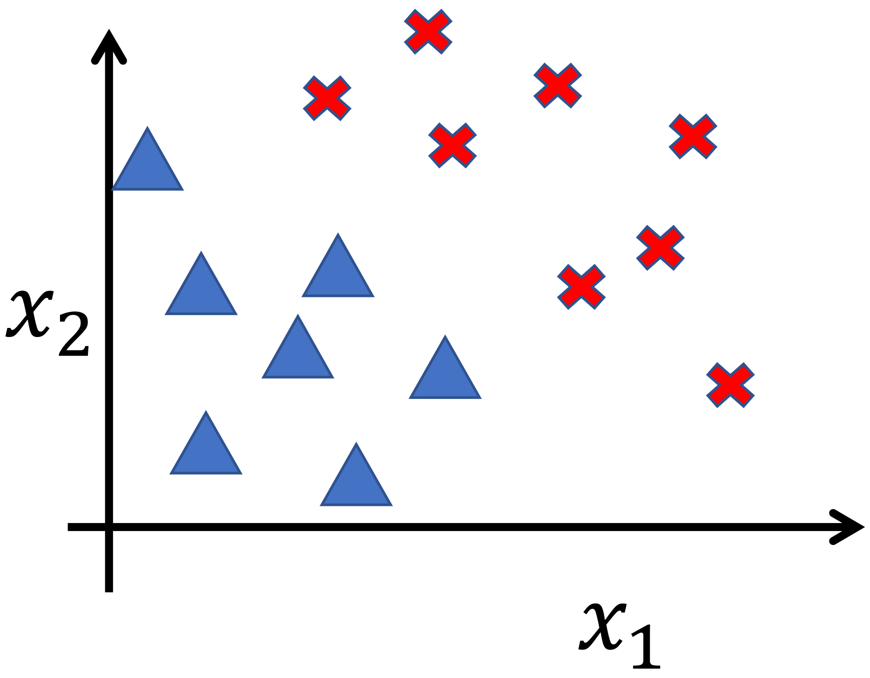
}

$$\frac{\partial}{\partial \theta_j} J(\theta)$$

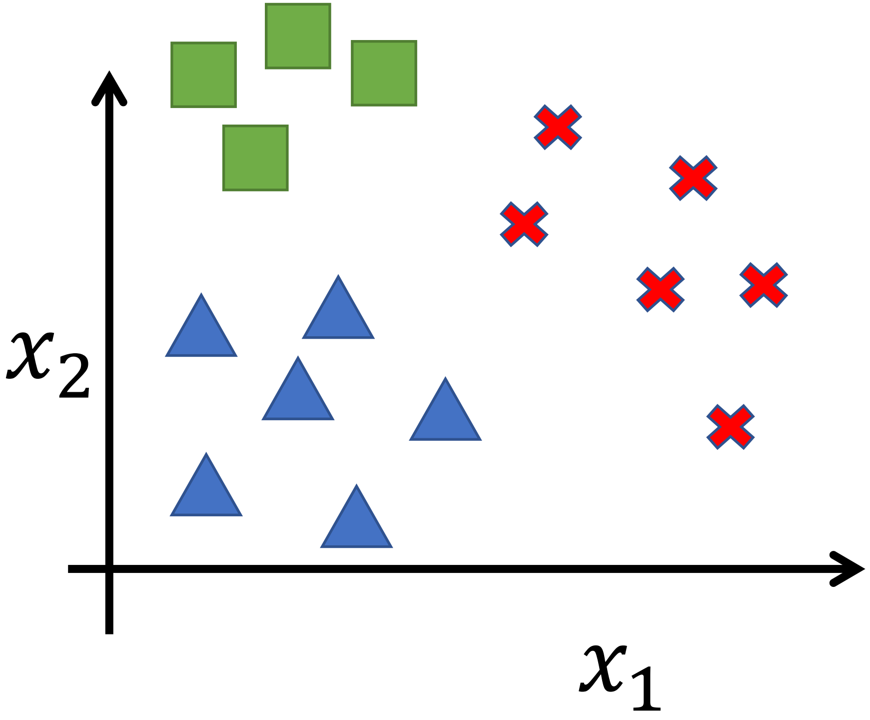
Multi-class classification

- Email foldering: Work, Friends, Family, Hobby
- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

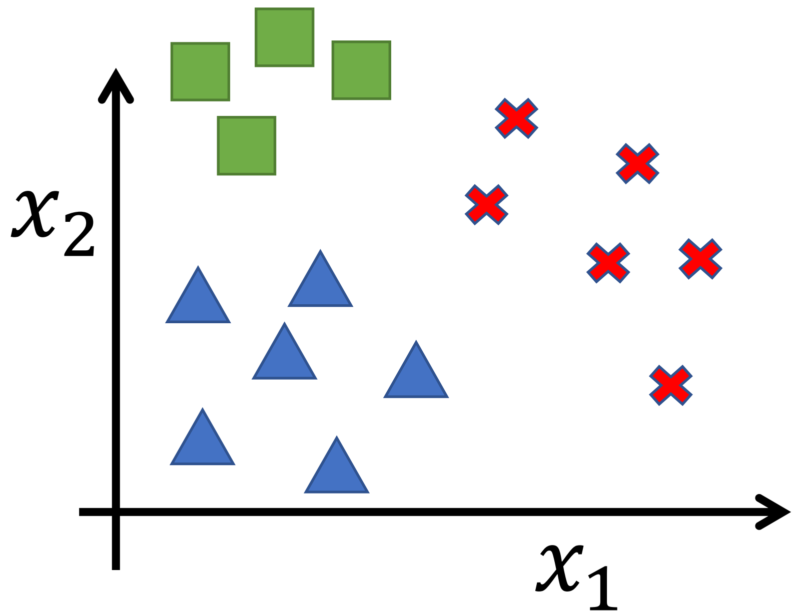
Binary classification

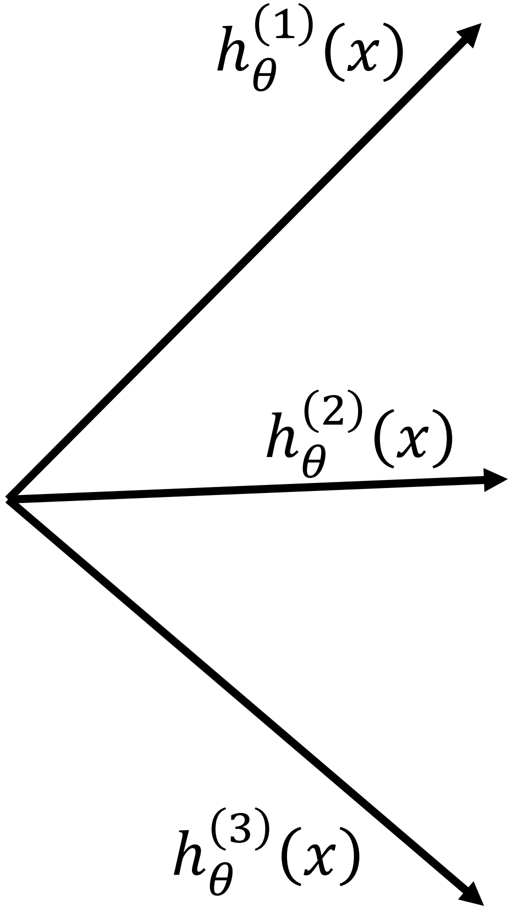


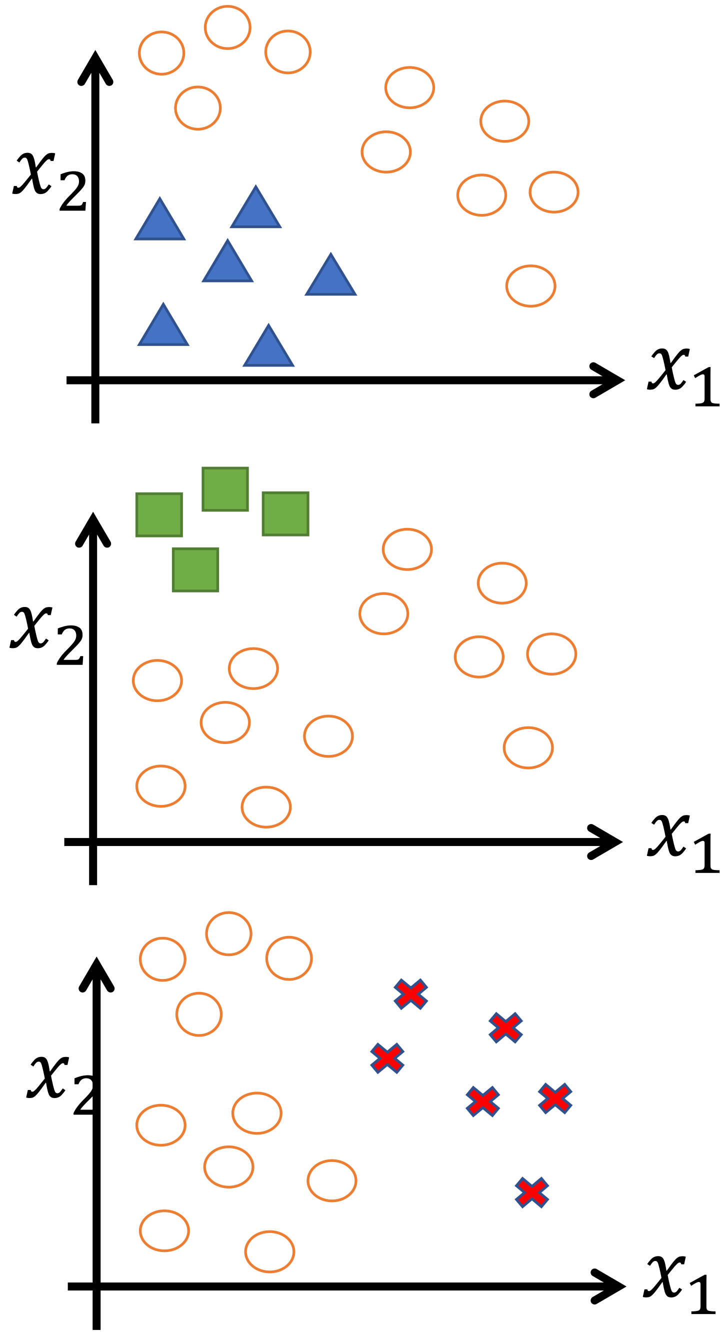
Multiclass classification



One-vs-all (one-vs-rest)







Class 1: ▲

Class 2: ■

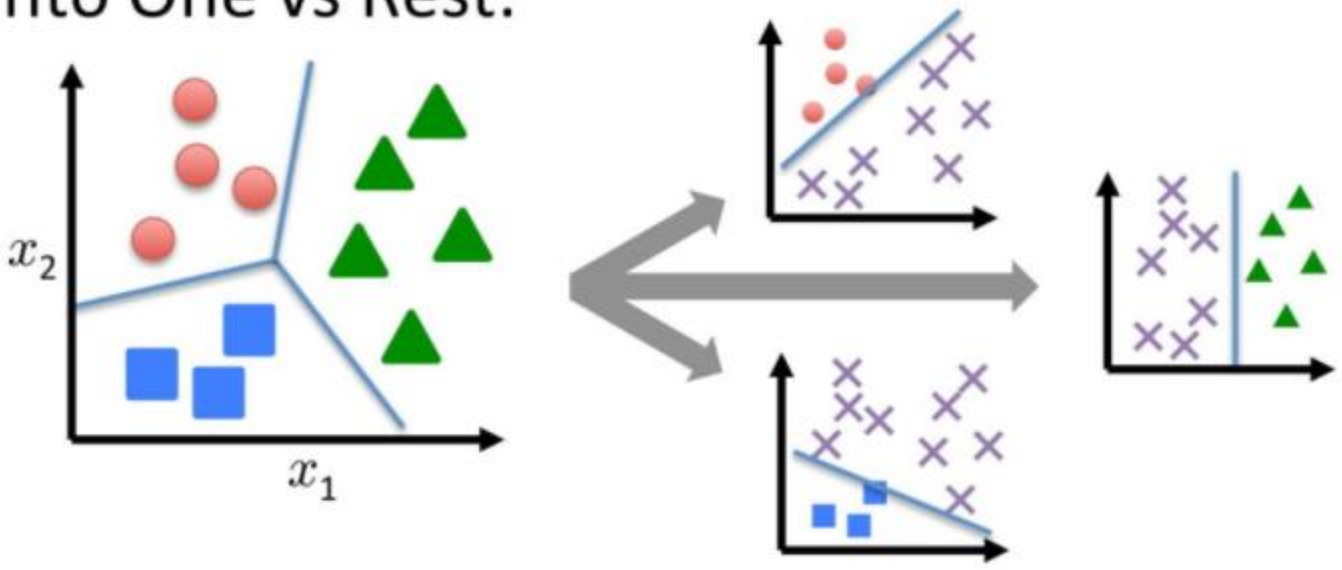
Class 3: ✖

$$h_\theta^{(i)}(x) = P(y = i | x; \theta) \quad (i = 1, 2, 3)$$

Softmax Function

Multi-Class Logistic Regression

Split into One vs Rest:



- Train a logistic regression classifier for each class i to predict the probability that $y = i$ with

$$h_c(\boldsymbol{x}) = \frac{\exp(\boldsymbol{\theta}_c^{\mathsf{T}} \boldsymbol{x})}{\sum_{c=1}^{C} \exp(\boldsymbol{\theta}_c^{\mathsf{T}} \boldsymbol{x})}$$

Implementing Multi-Class Logistic Regression

- Use $h_c(\boldsymbol{x}) = \dfrac{\exp(\boldsymbol{\theta}_c^\mathsf{T}\boldsymbol{x})}{\sum_{c=1}^{C}\exp(\boldsymbol{\theta}_c^\mathsf{T}\boldsymbol{x})}$ as the model for class c

- Gradient descent simultaneously updates all parameters for all models
  - Same derivative as before, just with the above $h_c(\boldsymbol{x})$

- Predict class label as the most probable label

$$\max_c h_c(\boldsymbol{x})$$

One-vs-all

- Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$

- Given a new input x, pick the class i that maximizes

$$\max_{\mathrm{i}} h_\theta^{(i)}(x)$$

⇒ For example, Three classes, cat, Dog, Horse.

so, $\hat{y} = \begin{bmatrix} 0.75 \\ 0.01 \\ 0.24 \end{bmatrix}$ → cat
→ Dog
→ Horse

↓

output of the model after applying softmax function

↳ All these values should be between $[0, 1]$

↳ And the total Prob = 1 → $[0.75 + 0.01 + 0.24] = 1$

References

- Machine Learning by Andrew Ng
- https://towardsdatascience.com/why-not-mse-as-a-loss-function-for-logistic-regression-589816b5e03c
- https://cs-114.org/wp-content/uploads/2021/02/5_LR_Jan_12_2021.pdf
- https://www.youtube.com/watch?v=Z8noL_0M4tw
- https://www.youtube.com/watch?v=z9XAXXGwUzM
- https://www.youtube.com/watch?v=xXvgkILaFT4