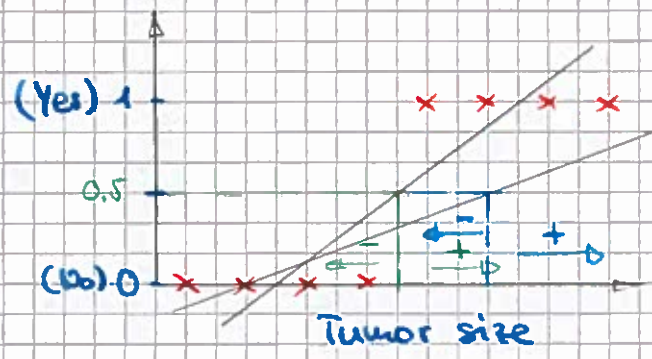
6. Logistic Regression

Classification:

EMail: Spam / not Spam?
Online transaction: fraudulent (Yes/No)?
Tumor: Malignant/Benign?

$y \in \{0,1\}$ 0: "Negative Class"
1: "Positive Class"

$y \in \{1,2,3,4\}$ multiclass classification Problem



Linear Regression is not useful for Classification Problems

Threshold classifier output $h_\theta(x)$ at 0.5

if $h_\theta(x) \geq 0.5$, predict 1
if $h_\theta(x) < 0.5$, predict 0

Classification: $y = 0$ or 1

$h_\theta(x)$ can be > 1 or < 0 Linear Regression

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

↓

Classification

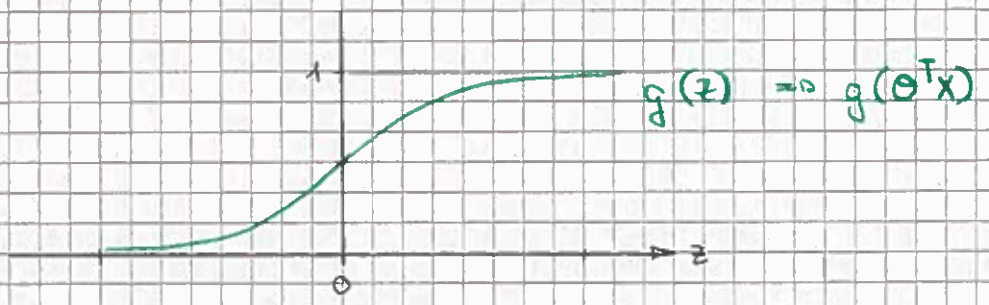
Logistic Regression Model

$\Rightarrow 0 \leq h_\theta(x) \leq 1$

$\Rightarrow h_\theta(x) = g(\theta^T x)$

mean the same: Sigmoid Function / Logistic Function

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



Interpretation of hypothesis output:

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

Example: if $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{Tumor size} \end{bmatrix}$

Tell patient that 70% chance of Tumor being malignant

$h_\theta(x) = p(y=1 \mid x; \theta)$ probability that $y=1$, given x, parameterized by θ.

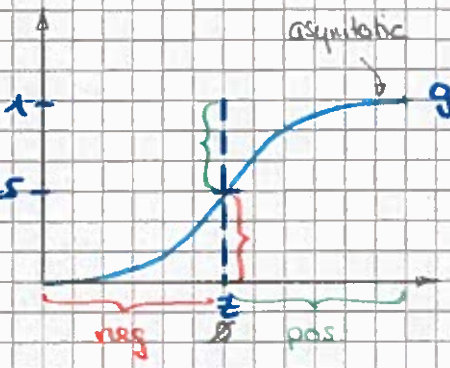
$y = 1$ or 0 $\Rightarrow P(y=0 \mid x;\theta) + P(y=1 \mid x;\theta) = 1$

$P(y=0 \mid x;\theta) = 1 - P(y=1 \mid x;\theta)$

Logistic Regression

$h_\theta(x) = g(\theta^T x) \Rightarrow p(y=1 \mid x; \theta)$

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



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

$\theta^T x \geq 0$

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

$h_\theta(x) = g(\theta^T x)$ $\quad g(z) < 0.5$

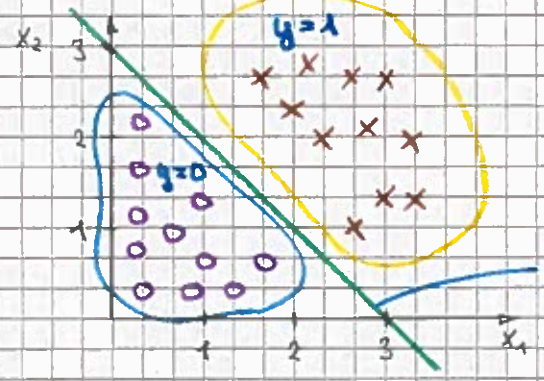
$\hookrightarrow \theta^T x < 0$

$g(z) \geq 0.5$ when $z \geq 0$

$h_\theta(x) = g(\theta^T x) \geq 0.5$ whenever $\theta^T x \geq 0$

z

Decision Boundary



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

$-3 \quad 1 \quad 1$

$\theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

Predict "$y=1$" if $\underbrace{-3 + x_1 + x_2}_{\theta^T x} \geq 0$ $\Rightarrow$ predicts $y=1$ if equation $-3 + x_1 + x_2 \geq 0$

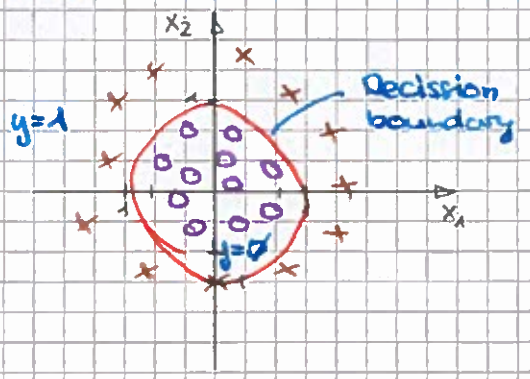
or $x_1 + x_2 \geq 3$ $\Rightarrow$

$\Downarrow$

corresponds to $h_\theta(x) = 0.5$

Decision Boundary is a property of the hypothesis and not of the training set.

Non-Linear decision boundaries



$$h_\theta(x) = g(\underset{-1}{\theta_0} + \underset{0}{\theta_1 x_1} + \underset{0}{\theta_2 x_2} + \underset{1}{\theta_3 x_1^2} + \underset{1}{\theta_4 x_2^2})$$

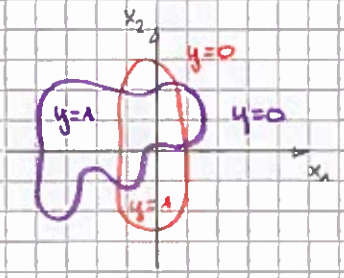
↳ $\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

predict "y=1" if $-1 + x_1^2 + x_2^2 \geq 0$

$$\boxed{x_1^2 + x_2^2 \geq 1}$$ ⇒ cycle

More complex boundaries

⇒ eg. $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \ldots)$



decision boundaries can take complex forms, depending on the polynomio ~~the~~ function.

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots (x^{(m)}, y^{(m)})\}$

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

$\mathbb{R}^{n+1}$

hypothesis: $h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$ ⇒ based on param θ.

⇒ linear Regression uses the following function to determine θ.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{2} \cdot (h_\theta(x)^{(i)} - y^{(i)})^2}$$

⇓ replace resp. substitute squared error function bei cost()

$$\text{cost}(h_\theta(x^{(i)}), y) = \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$$

⇓ evaluates the cost for individual example as used in linear Regr

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

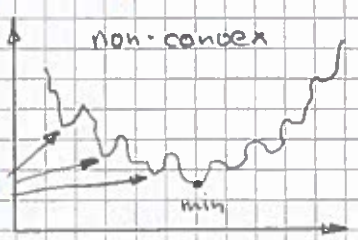
↳ this is now the same as linear regression

⇓

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

If this function is used in logistic regression ⇒ this is a non-convex function for parameter optimization.

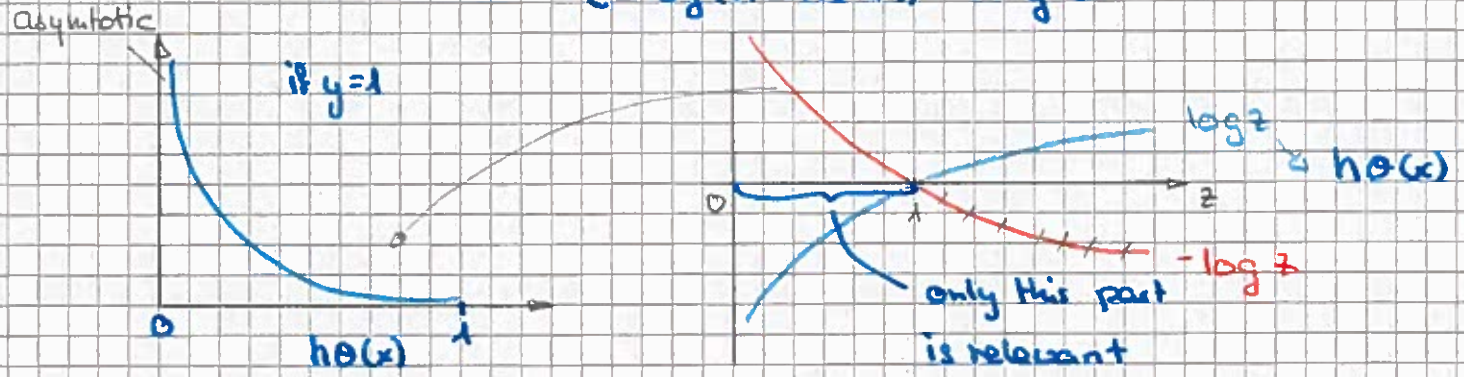
⇒ Could work, but usually does not.



function has multiple local minimums

Logistic regression cost function for $y=1$

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

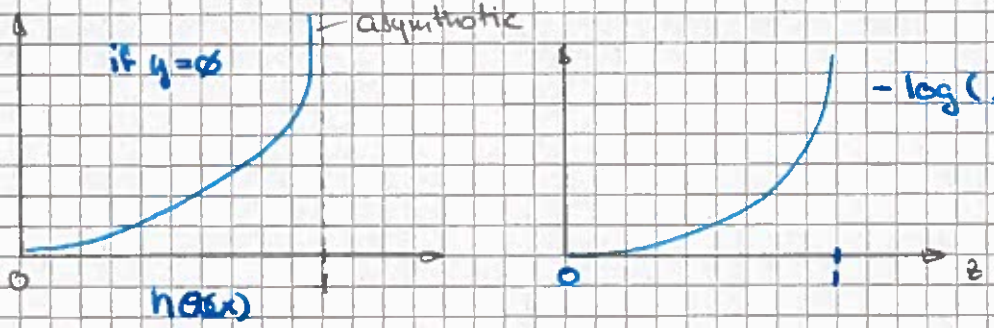


⇒ Cost = 0 if $y=1$, $h_\theta(x)=1$
But as $h_\theta(x) \to 0$ ⇒ Cost $\to \infty$

⇒ Captures intuition that if $h_\theta(x)=0$,
predict $P(y=1 \mid x;\theta)=0$, but $y=1$
we will penalize learning algorithm by very large cost.

Logistic regression cost function for $y=0$

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



- ☑ if $h_\theta(x)=y$, then $\text{cost}(h_\theta(x), y)=0$ (for $y=0$ and $y=1$)
- ☑ if $y=0$ then $\text{cost}(h_\theta(x), y) \to \infty$ as $h_\theta(x) \to 1$
- ☑ Regardless of wheather $y=0$ or $y=1$, if $h_\theta(x)=0.5$, then $\text{Cost}(h_\theta(x), y) > 0$

Logistic regression cost function

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

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

Create a simpler cost function by aggregating both functions

Note: y is always 0 or 1

$$Cost(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y) \log(1 - h_\theta(x))$$

($y = 0$: $= 0 \cdot \log(\dots) = 0$; $y = 1$: $= -\log(\dots)$; $1 - y = 0$: $= -\log(\dots)$; $1 - y = 1$: $= 0 \cdot \log(\dots) = 0$)

if $y = 1$ $Cost(h_\theta(x), y) = -\log h_\theta(x)$

if $y = 0$ $Cost(h_\theta(x), y) = -\log(1 - h_\theta(x))$

$\Downarrow$

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

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

maximum likelyhood estimation

widely used cost function for logistic regression problems

It has the good property of being convex!

to fit parameters θ:

$$\min_\theta J(\theta) \Rightarrow \text{get } \theta$$

to make a prediction given new x:

$$\text{Output } h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \Rightarrow P(y = 1 \mid x; \theta)$$

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

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

Repeat {

$$\theta_j := \theta_j - \alpha \underbrace{\frac{\partial}{\partial \theta_j} \cdot J(\theta)}$$

}

→ partial derivative

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \Rightarrow \text{error}$$

$$\Downarrow$$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \Rightarrow \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

Algorithm looks identical to linear regression

for linear regression : $h\theta(x) = \theta^T x$

for logistic regression: $h\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

Feature scaling also applies to logistic regression !

Optimization algorithm

Given θ, we have code that can compute

- $J(\theta)$
- $\frac{\partial}{\partial \theta_j} J(\theta)$ for $(j = 0, 1, \ldots, n)$

Optimization algorithms:

- gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

} more advanced algorithms (with a line search to define α)

Advantages:
- no need to manually pick α
- often faster than gradient descent

Disadvantage:
- more complex

Example:

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \Rightarrow \min_{\theta} J(\theta)$$

$\theta_1 = 5 ; \theta_2 = 5$

$J(\theta) = (\theta_1 - 5)^2 + (\theta_2 - 5)^2$

$\frac{\partial}{\partial \theta_1} = 2(\theta_1 - 5)$

$\frac{\partial}{\partial \theta_2} = 2(\theta_2 - 5)$

```
function [jval, gradient] = costFunction(theta)
  jval = (theta(1)-5)^2 +
         (theta(2)-5)^2;
  gradient = zeros(2,1);
  gradient(1) = 2*(theta(1)-5)
  gradient(2) = 2*(theta(2)-5)
options = optimset('GradObj','on',
                   'MaxIter','100');
initialTheta = zeros(2,1)
[optTheta, functionVal, exitFlag] ...
= fminunc(@costFunction, initialTheta, options)
```

$\theta \in \mathbb{R}^d \quad d \geq 2$

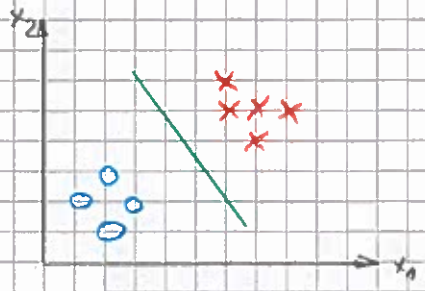
Must be at least 2 dim. vector

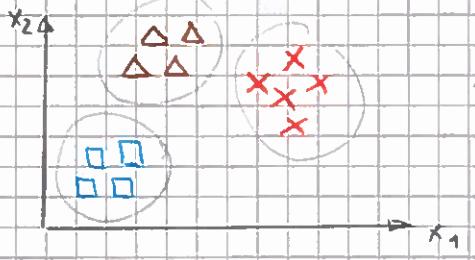
Multiclass Classification

- eMail Foldering / tagging: Work, Friends, Family, Hobby
  $y=1$ $y=2$ $y=3$ $y=4$

- Medical diagrams: Not ill, cold, flu
  $y=1$ 2 3

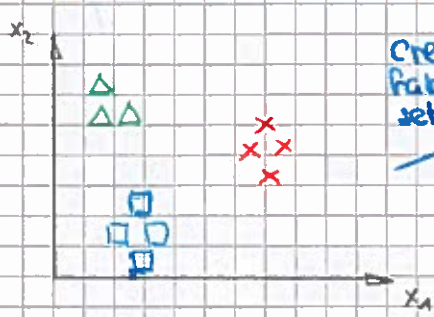
- Weather: Sunny, Cloudy, Rain, Snow
  $y=1$ 2 3 4





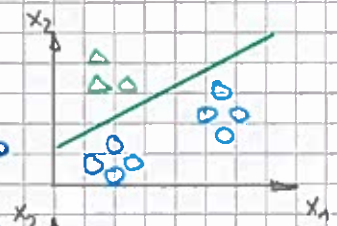
3 different classes

One vs. all (One vs. rest)

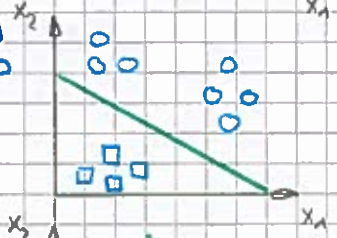


Create new fake training set

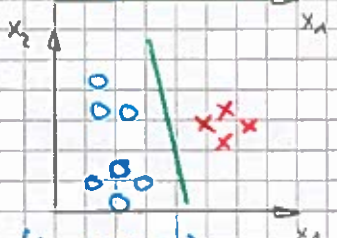
$\triangle \Rightarrow y=1$
$\circ \Rightarrow y=0$



$h_\theta^{(1)}(x)$
$\Rightarrow P(y=1 \mid x;\theta)$



$h_\theta^{(2)}(x)$



$h_\theta^{(3)}(x)$

Class 1: $\triangle$
Class 2: $\times$
Class 3: $\square$

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

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

On a new input x, to make a prediction, pick the class i that maximizes. $\Rightarrow$ highest probability

$\max_i h_\theta^{(i)}(x)$ $\Rightarrow$ which gives highest probability