



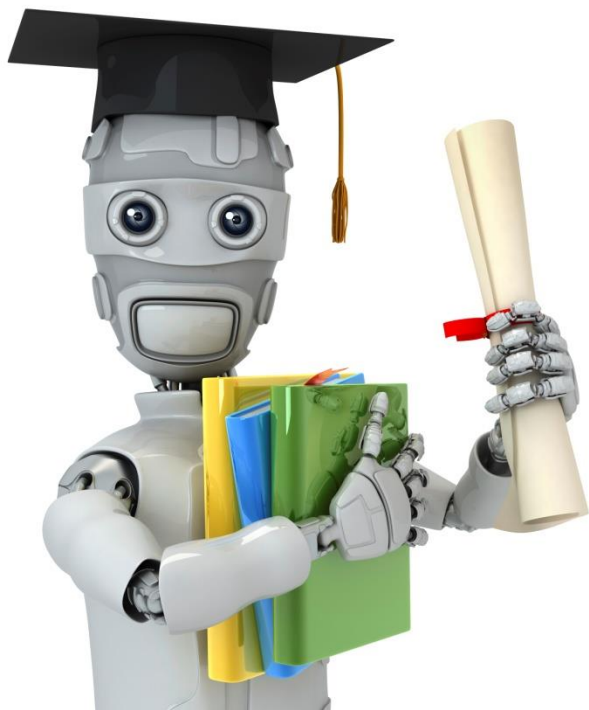
# CSE 4621

# Machine Learning

Lecture 5

**Md. Hasanul Kabir, PhD.**  
Professor, CSE Department  
Islamic University of Technology (IUT)





# Logistic Regression

---

# Classification

Machine Learning

**Source & Special Thanks to Andrew Ng (Coursera) Machine Learning Course**

# Classification

Email: Spam / Not Spam?

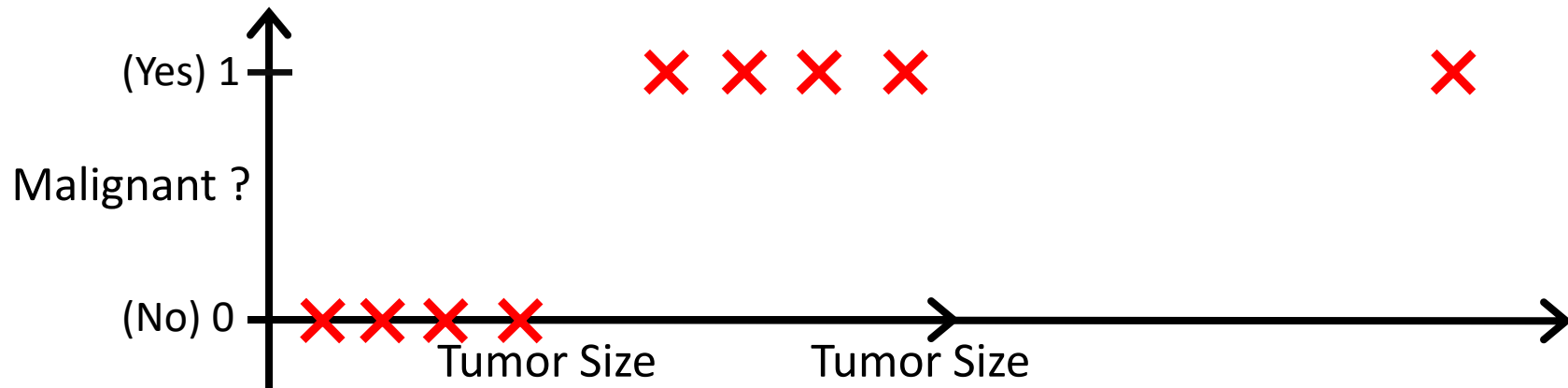
Online Transactions: Fraudulent (Yes / No)?

Tumor: Malignant / Benign ?

$$y \in \{0, 1\}$$

0: “Negative Class” (e.g., benign tumor)

1: “Positive Class” (e.g., malignant tumor)



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

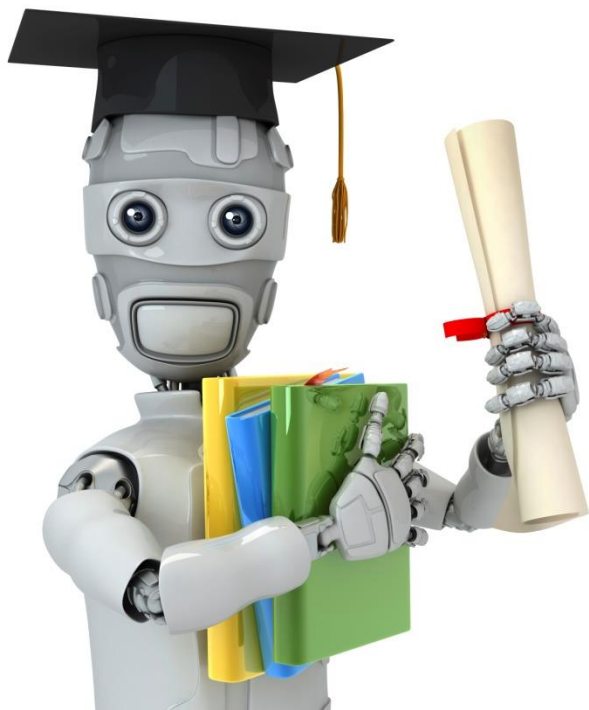
If  $h_{\theta}(x) \geq 0.5$ , predict "y = 1"

If  $h_{\theta}(x) < 0.5$ , predict "y = 0"

Classification:  $y = 0$  or  $1$

$h_{\theta}(x)$  can be  $> 1$  or  $< 0$

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



Machine Learning

# Logistic Regression

---

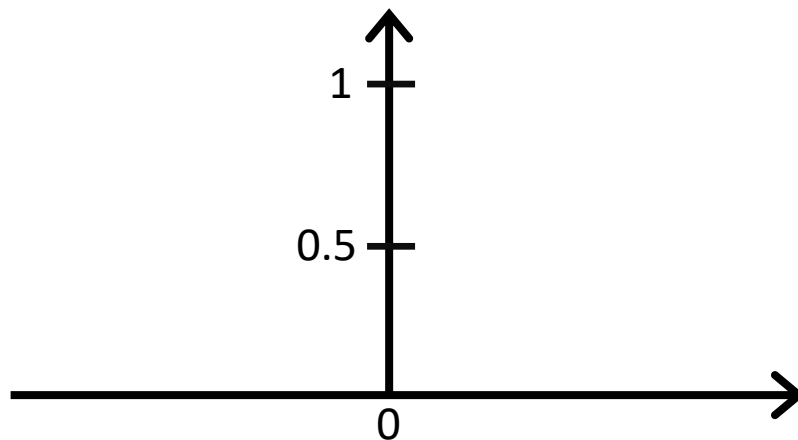
## Hypothesis Representation

## Logistic Regression Model

Want  $0 \leq h_{\theta}(x) \leq 1$

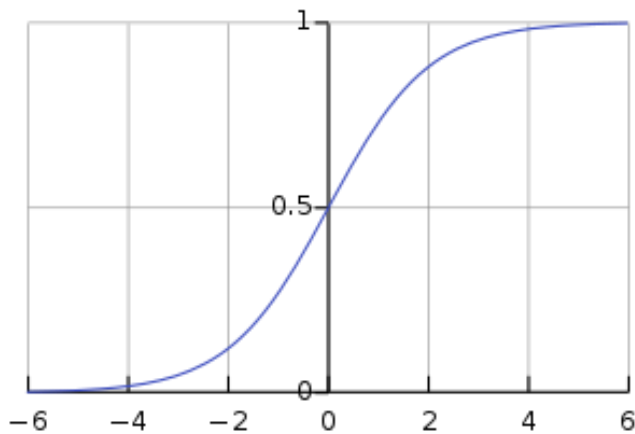
$$h_{\theta}(x) = \theta^T x$$

Sigmoid function  
Logistic function

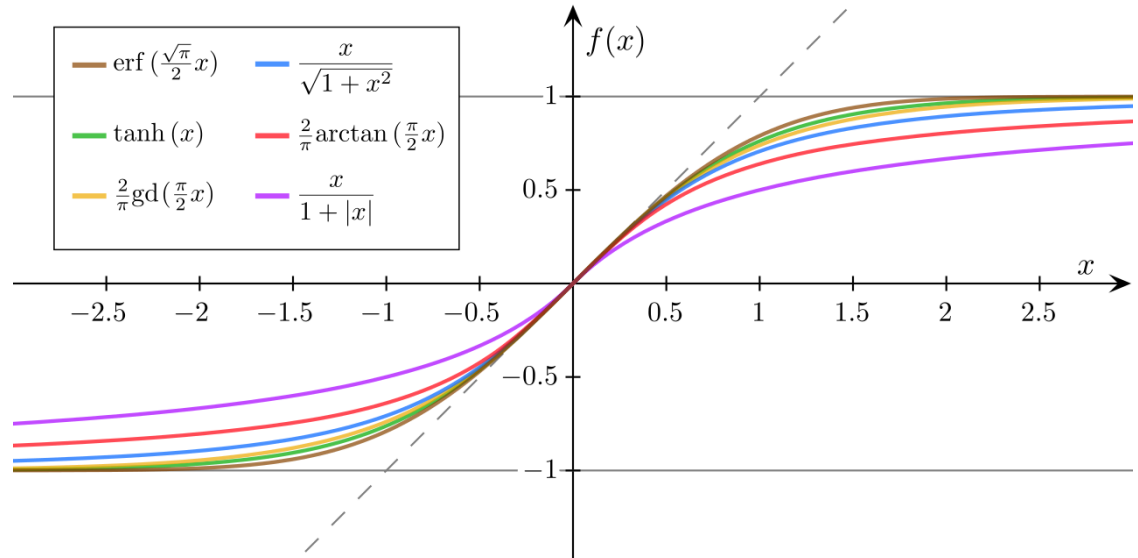


# Sigmoid Functions

- Having a characteristic "S"-shaped curve or **sigmoid curve**.



Logistic Function



Other Examples of Sigmoid Functions



## Interpretation of Hypothesis Output

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

Example: If  $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

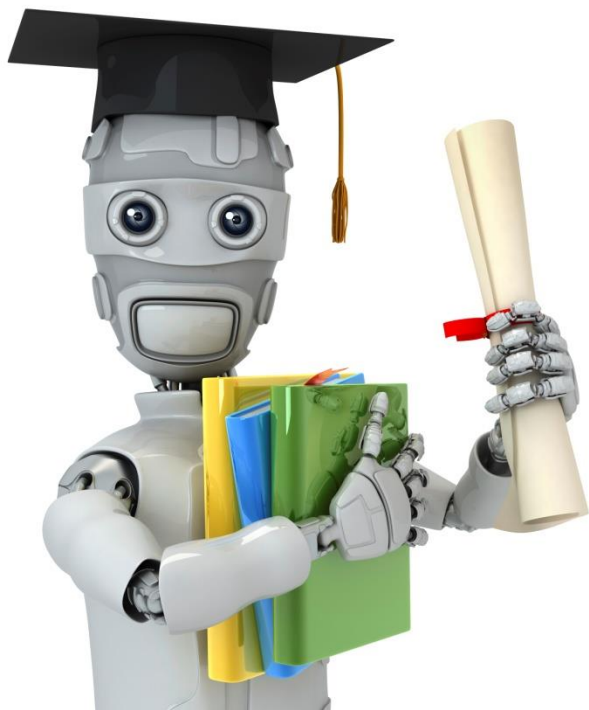
$$h_{\theta}(x) = 0.7$$

Tell patient that 70% chance of tumor being malignant

“probability that  $y = 1$ , given  $x$ ,  
parameterized by  $\theta$ ”

$$P(y = 0|x; \theta) + P(y = 1|x; \theta) = 1$$

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



Machine Learning

# Logistic Regression

---

## Decision boundary

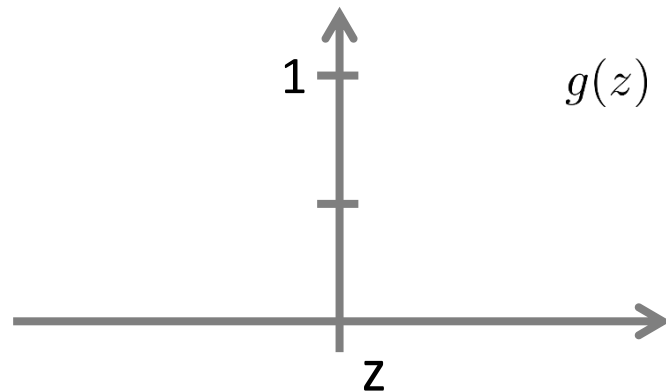
## Logistic regression

$$h_{\theta}(x) = g(\theta^T x)$$

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

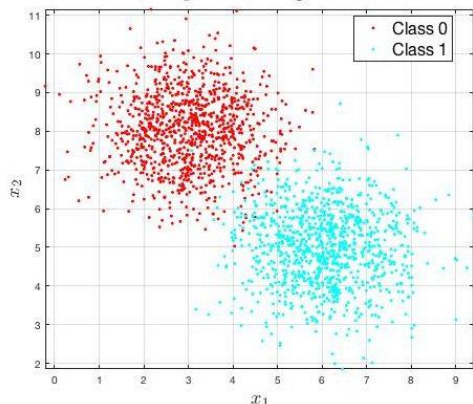
Suppose predict “ $y = 1$ ” if  $h_{\theta}(x) \geq 0.5$

predict “ $y = 0$ ” if  $h_{\theta}(x) < 0.5$

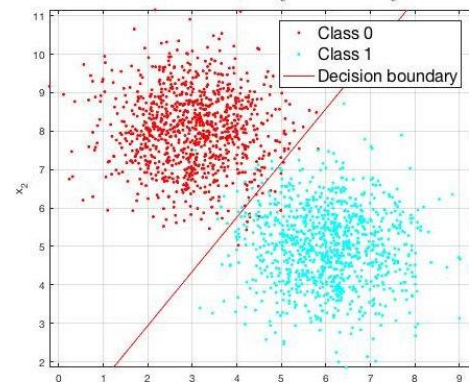


# Logistic Function

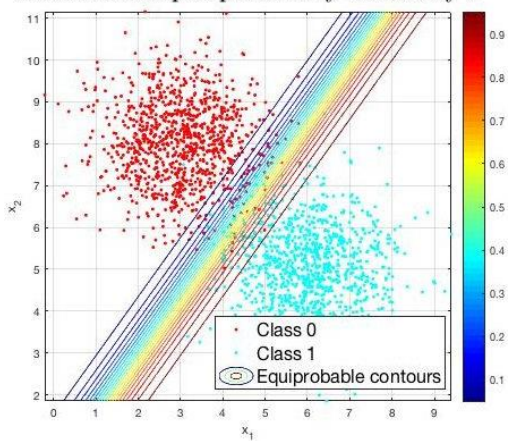
Input training data



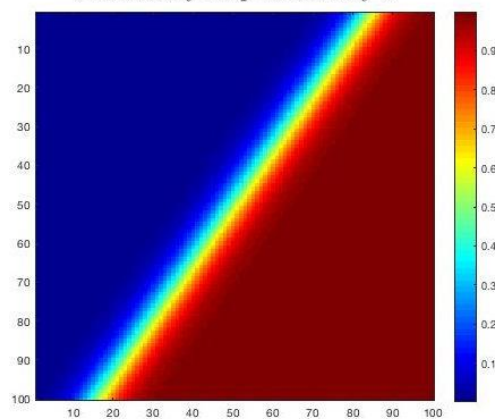
Decision boundary defined by  $\theta$



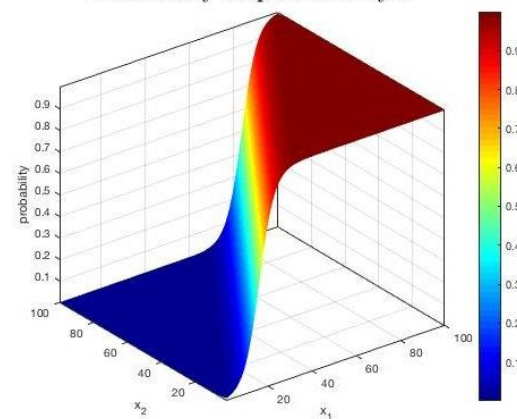
Contours of equal probability defined by  $\theta$



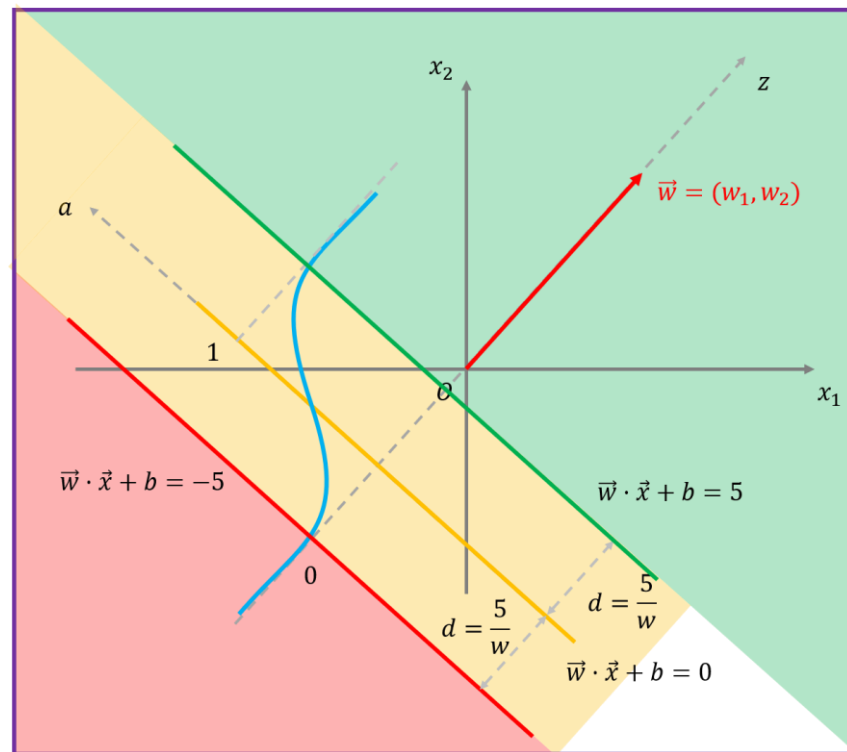
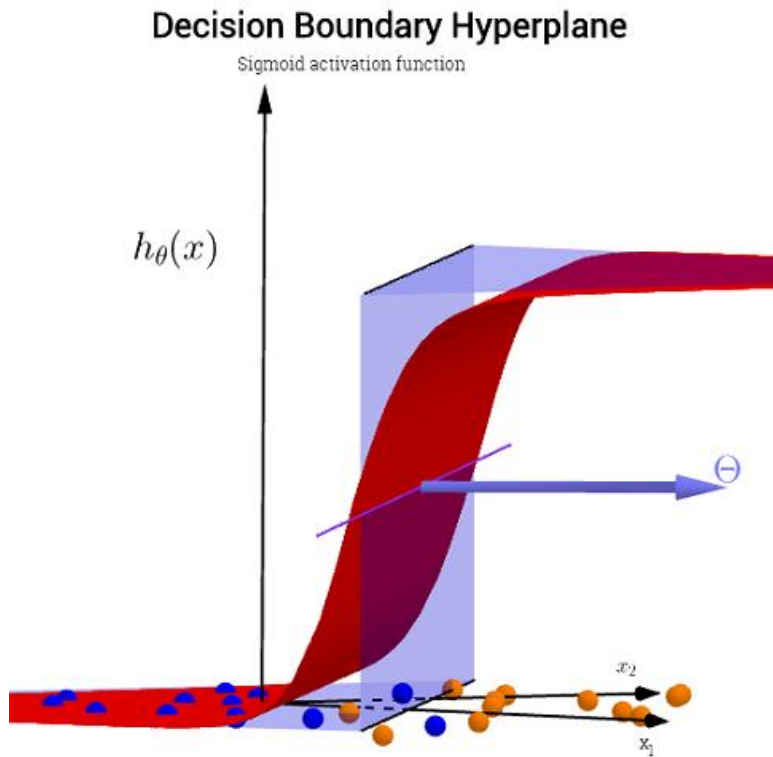
Probability map defined by  $\theta$



Probability map defined by  $\theta$



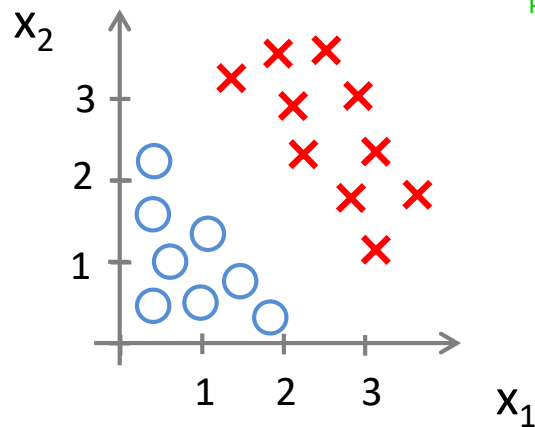
# Logistic Function



# Decision Boundary

How to fit (find) Parameter  $\theta$

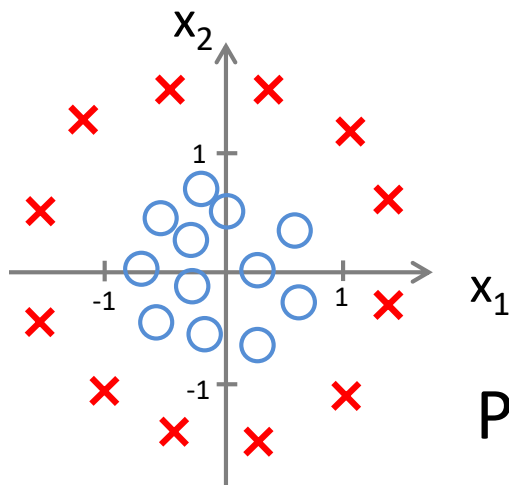
Parameter  $\theta$  ( $\theta_0$ ,  $\theta_1$ ,  $\theta_2$ ) defines the decision boundary  
not the training set. Training set may be used to find the  
Parameter  $\theta$



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

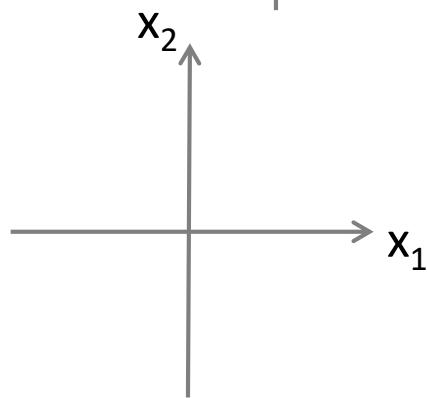
Predict “ $y = 1$ ” if  $-3 + x_1 + x_2 \geq 0$

## Non-linear decision boundaries

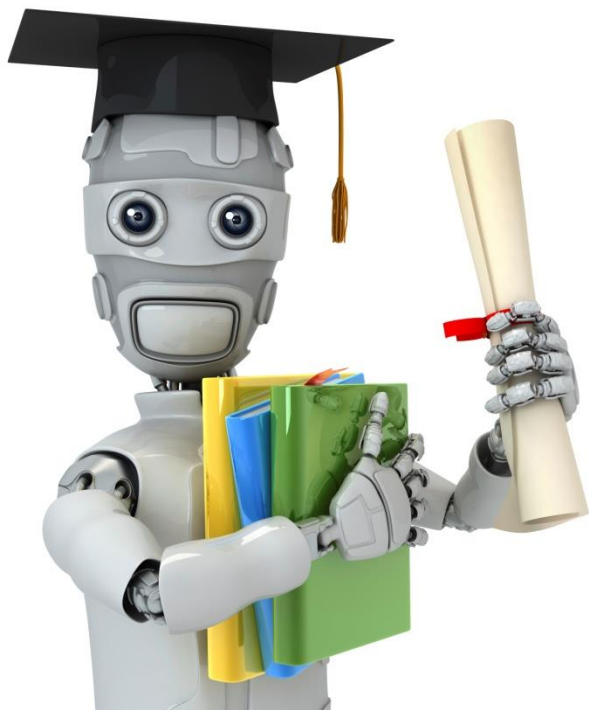


$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Predict “ $y = 1$ ” if  $-1 + x_1^2 + x_2^2 \geq 0$



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



Machine Learning

# Logistic Regression

---

## Cost function



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

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

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

How to choose parameters  $\theta$  ?

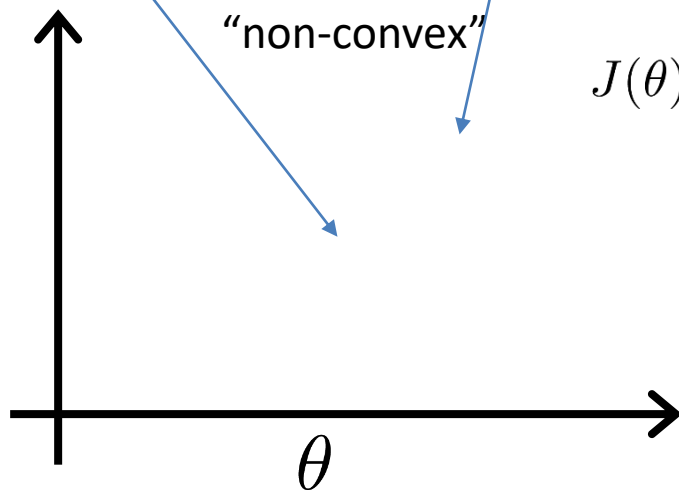
# Cost function

$J(\theta)$  is non-linear because of the presence of non-linear sigmoid function

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

Non-Linear Function

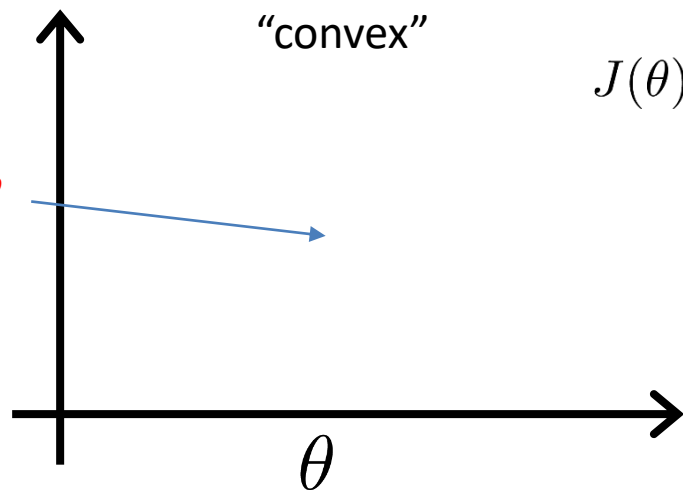
$$\text{Cost}(h_{\theta}(x^{(i)}), y^{(i)}) = \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2$$



"non-convex"

$J(\theta)$

We want  $J(\theta)$  to behave like this



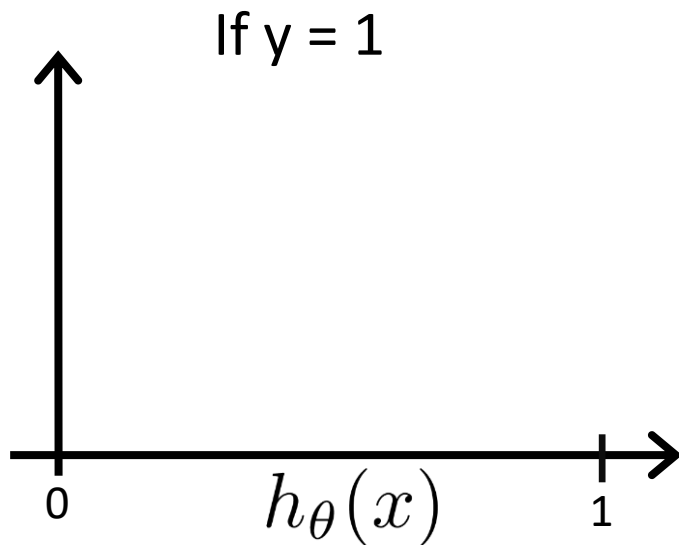
"convex"

$J(\theta)$

# Logistic regression cost function

*Different  
Cost Function*

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



Cost = 0 if  $y = 1, h_{\theta}(x) = 1$

But as  $h_{\theta}(x) \rightarrow 0$

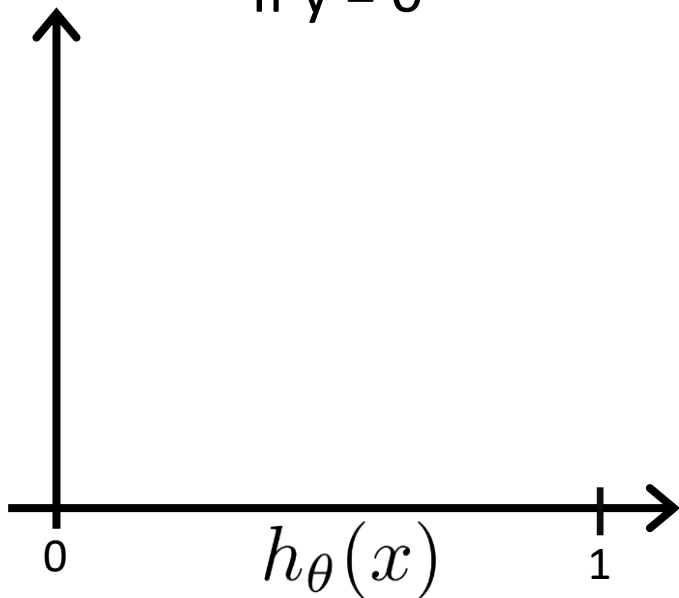
$\text{Cost} \rightarrow \infty$

Captures intuition that if  $h_{\theta}(x) = 0$ ,  
(predict  $P(y = 1|x; \theta) = 0$ ), but  $y = 1$ ,  
we'll penalize learning algorithm by a very  
large cost.

## Logistic regression cost function

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

If  $y = 0$



In logistic regression, the cost function for our hypothesis outputting (predicting)  $h_\theta(x)$  on a training example that has label  $y \in \{0, 1\}$  is:

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

Which of the following are true? Check all that apply.

If  $h_\theta(x) = y$ , then  $\text{cost}(h_\theta(x), y) = 0$  (for  $y = 0$  and  $y = 1$ ).

Well done!

If  $y = 0$ , then  $\text{cost}(h_\theta(x), y) \rightarrow \infty$  as  $h_\theta(x) \rightarrow 1$ .

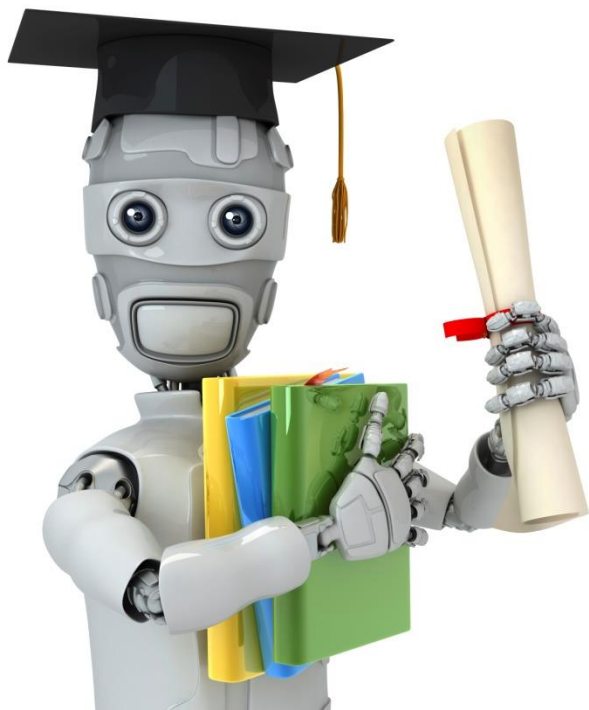
Well done!

If  $y = 0$ , then  $\text{cost}(h_\theta(x), y) \rightarrow \infty$  as  $h_\theta(x) \rightarrow 0$ .

Well done!

Regardless of whether  $y = 0$  or  $y = 1$ , if  $h_\theta(x) = 0.5$ , then  $\text{cost}(h_\theta(x), y) > 0$ .

Well done!



Machine Learning

# Logistic Regression

---

Simplified cost function  
and gradient descent

## Logistic regression cost function

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

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

Note:  $y = 0$  or  $1$  always

## Logistic regression cost function

$x^{(i)}$  = input (features) of  $i^{th}$  training example.

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

$x_j^{(i)}$  = value of feature  $j$  in  $i^{th}$  training example

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

To fit parameters  $\theta$ :

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

To make a prediction given new  $x$ :

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



## Gradient Descent

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

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

Repeat {

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

} (simultaneously update all  $\theta_j$ )

## Gradient Descent

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

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

Repeat {

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

} (simultaneously update all  $\theta_j$ )

Algorithm looks identical to linear regression!

## Optimization algorithm

Cost function  $J(\theta)$ . Want  $\min_{\theta} J(\theta)$ .

Given  $\theta$ , we have code that can compute

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

Gradient descent:

Repeat {

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

}

# Optimization algorithm

Given  $\theta$ , we have code that can compute

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

Optimization algorithms:

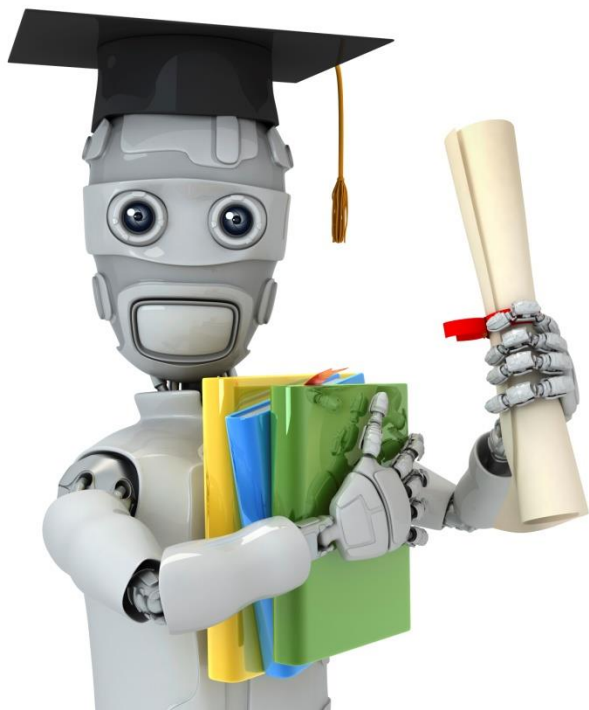
- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

Advantages:

- No need to manually pick  $\alpha$
- Often faster than gradient descent.

Disadvantages:

- More complex



Machine Learning

# Logistic Regression

---

Multi-class classification:  
One-vs-all

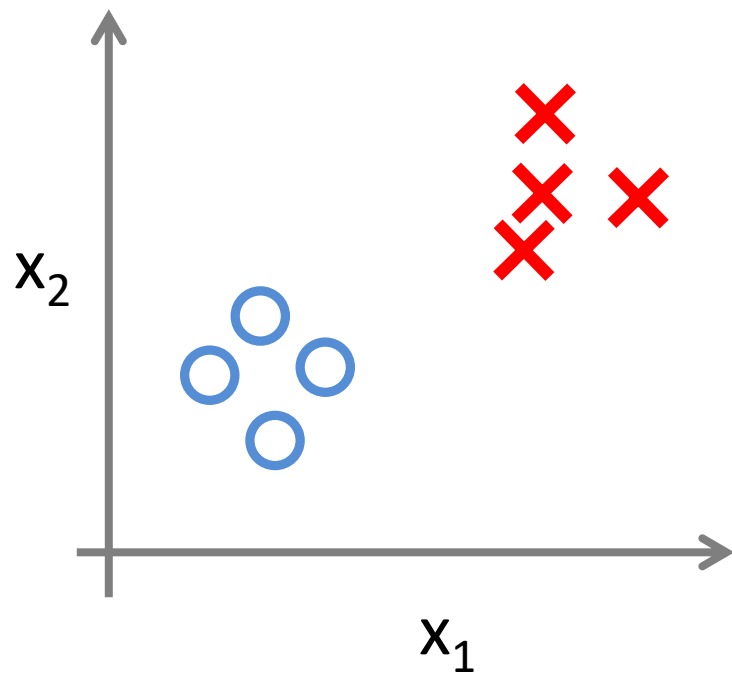
## **Multiclass classification**

Email foldering/tagging: Work, Friends, Family, Hobby

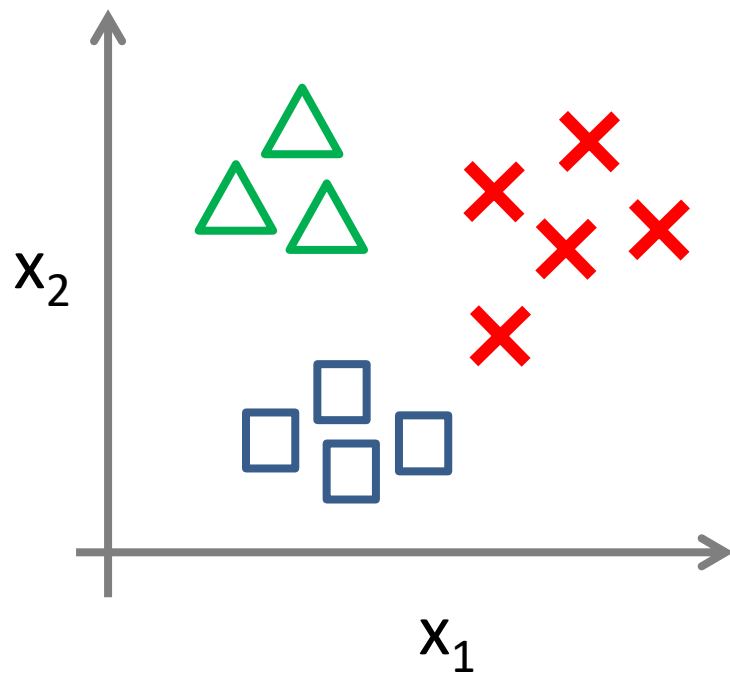
Medical Diagnosis: Not ill, Cold, Flu

Weather State Prediction: Sunny, Cloudy, Rain, Snow

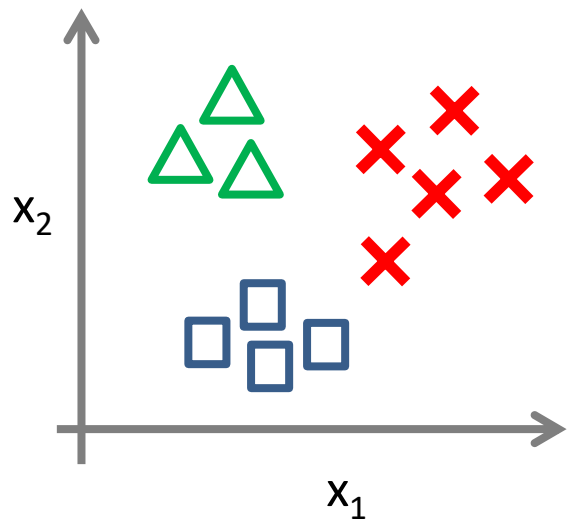
Binary classification:





Multi-class classification:



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

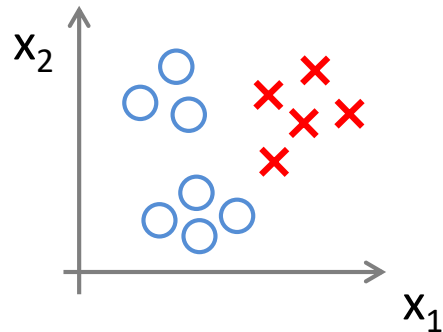
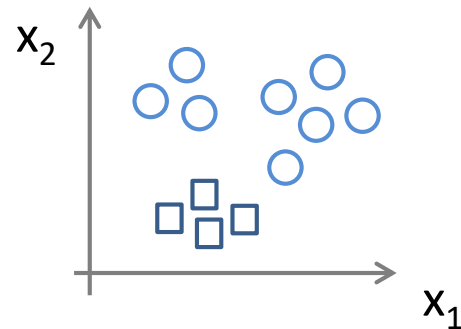
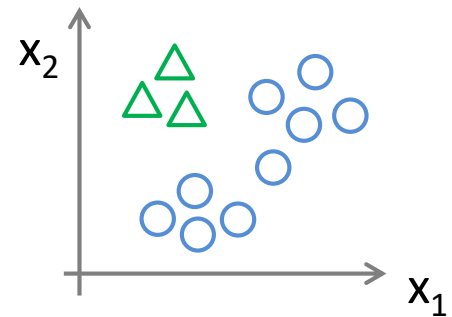


Class 1: 

Class 2: 

Class 3: 

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





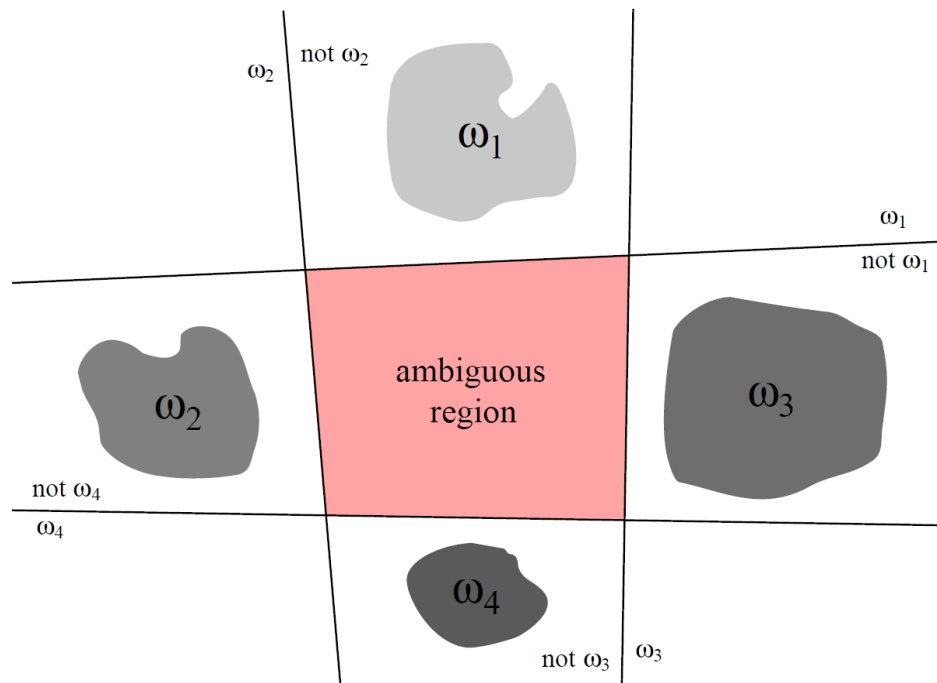
## One-vs-all

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

On a new input  $x$ , to make a prediction, pick the class  $i$  that maximizes

$$\max_i h_{\theta}^{(i)}(x)$$

# One-vs-All



# One-vs-One

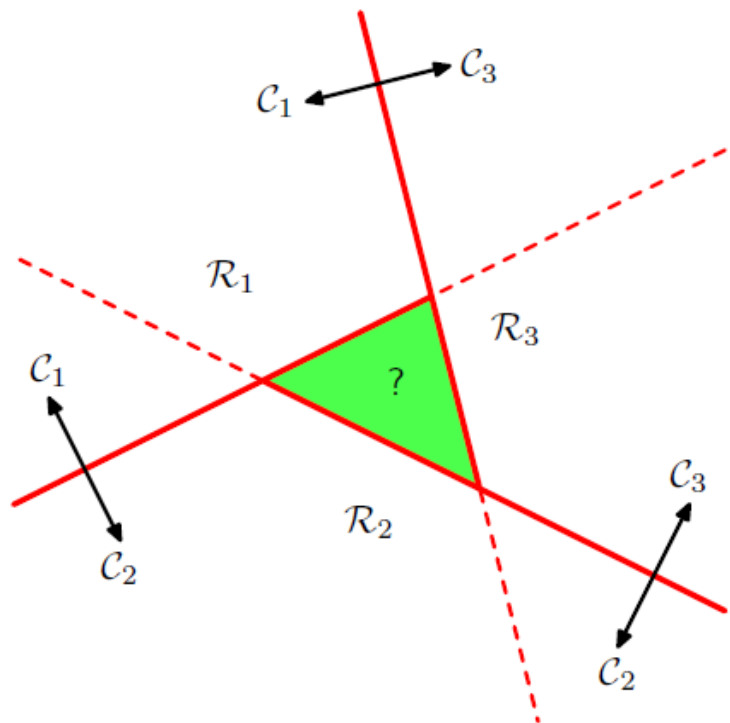


Fig: 3-class problem

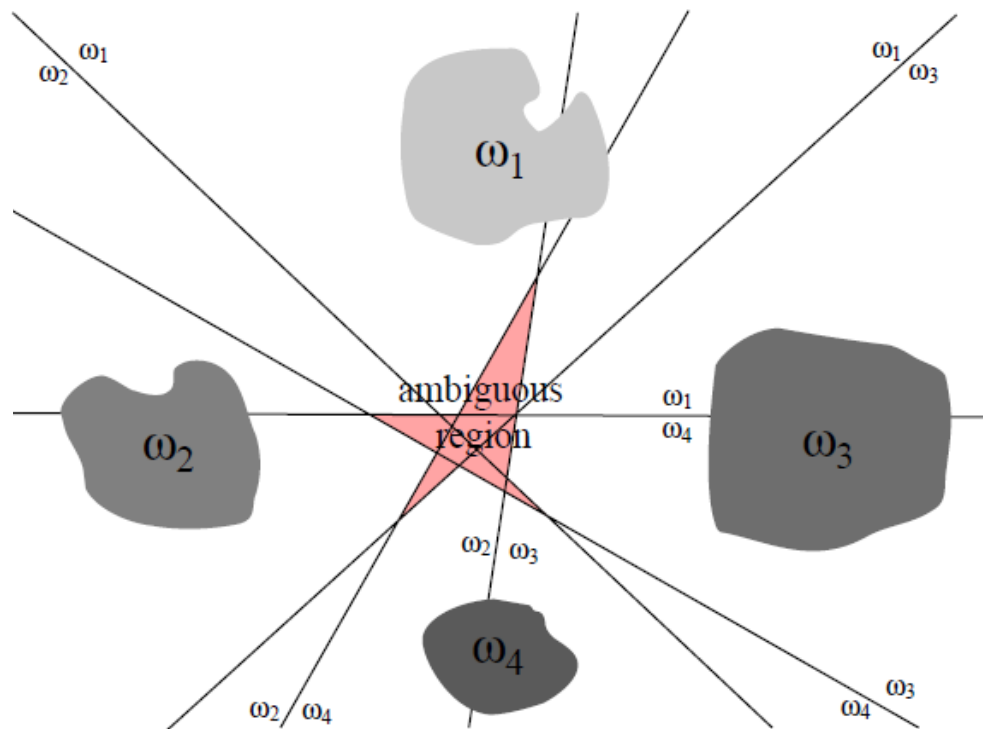


Fig: 4-class problem

# Linear Machine

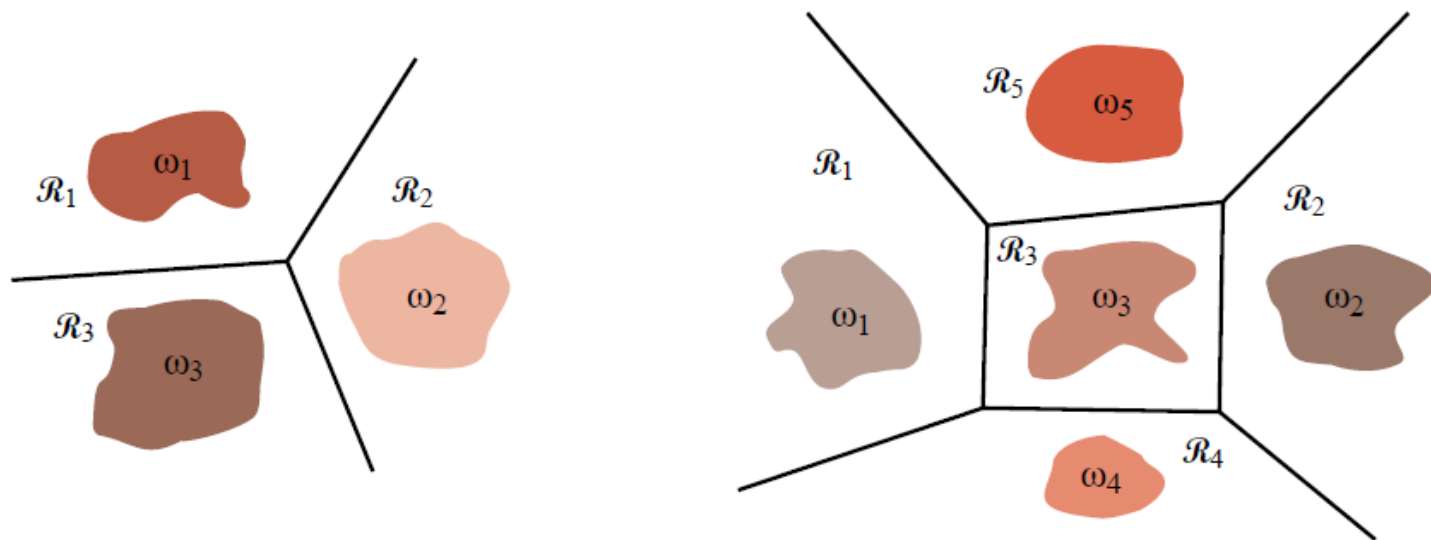


Figure 5.4: Decision boundaries produced by a linear machine for a three-class problem and a five-class problem.