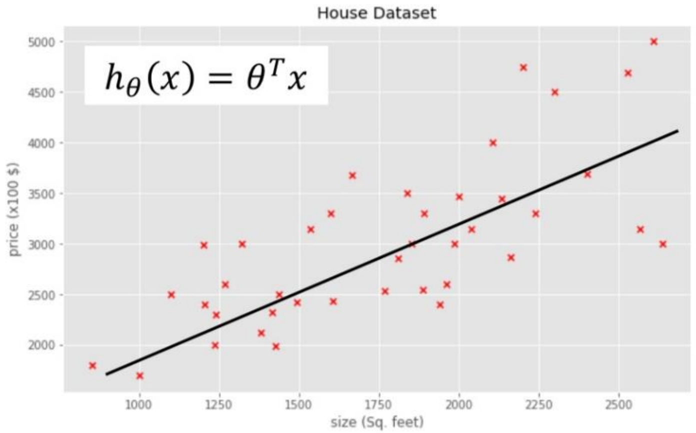
Machine Learning

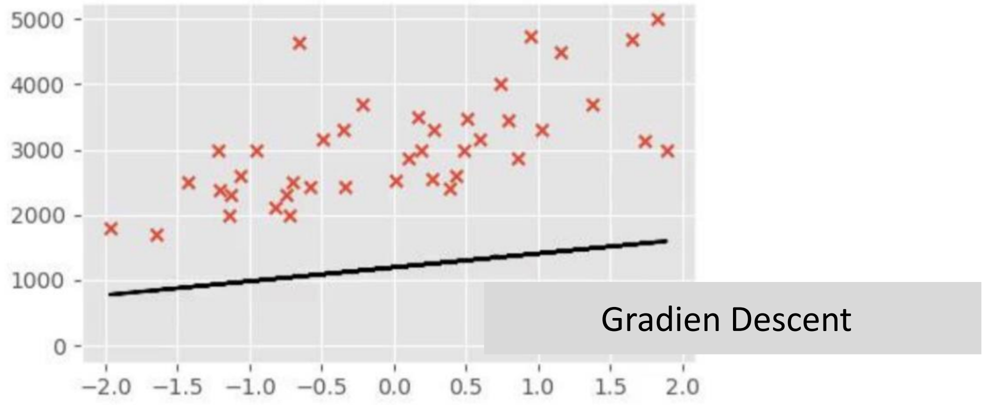
By Ghazal Lalooha

Classification: logistic regression

Reminder: Regression

- Target. Estimation of a continuous quantity according to attribute values.





Cost function

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

minizing

Classification

Email: Spam (yes/no?)
Online Transaction: Fraud (Yes / No?)
Cancerous gland: benign / malignant?

In these examples, the variable whose value we want to predict has two values:

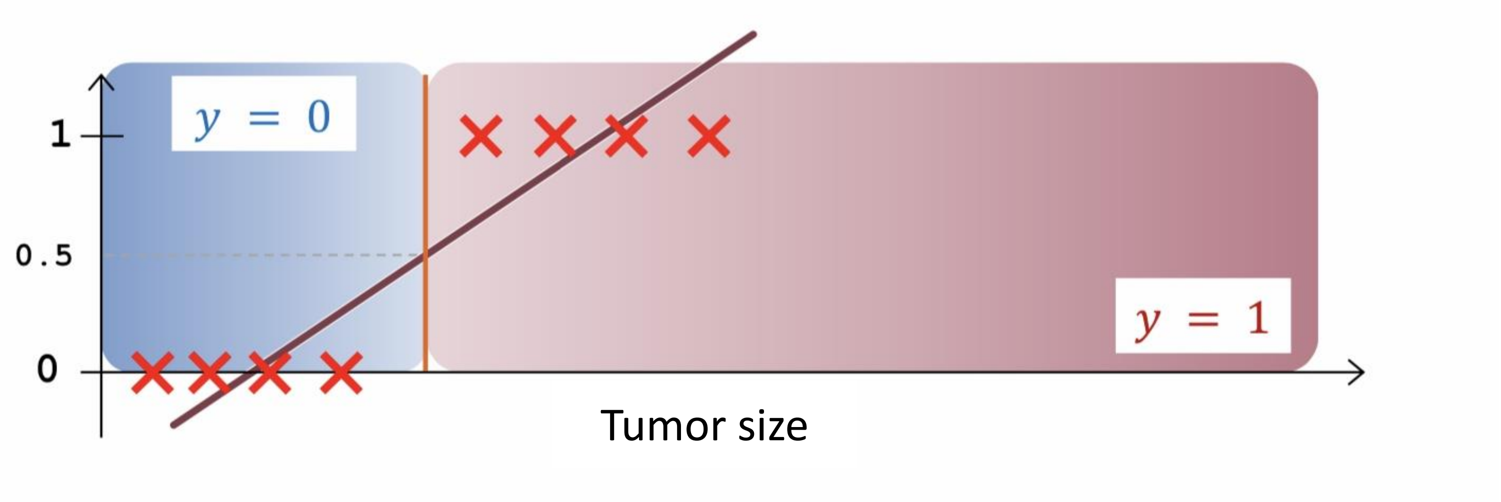
$y \in \{0,1\}$	Zero: "Negative class" (like benign tumor) One: "positive class" (such as malignant tumor)

Classification: predicting a variable with discrete values.

- Binary classification
- Multiclass classification

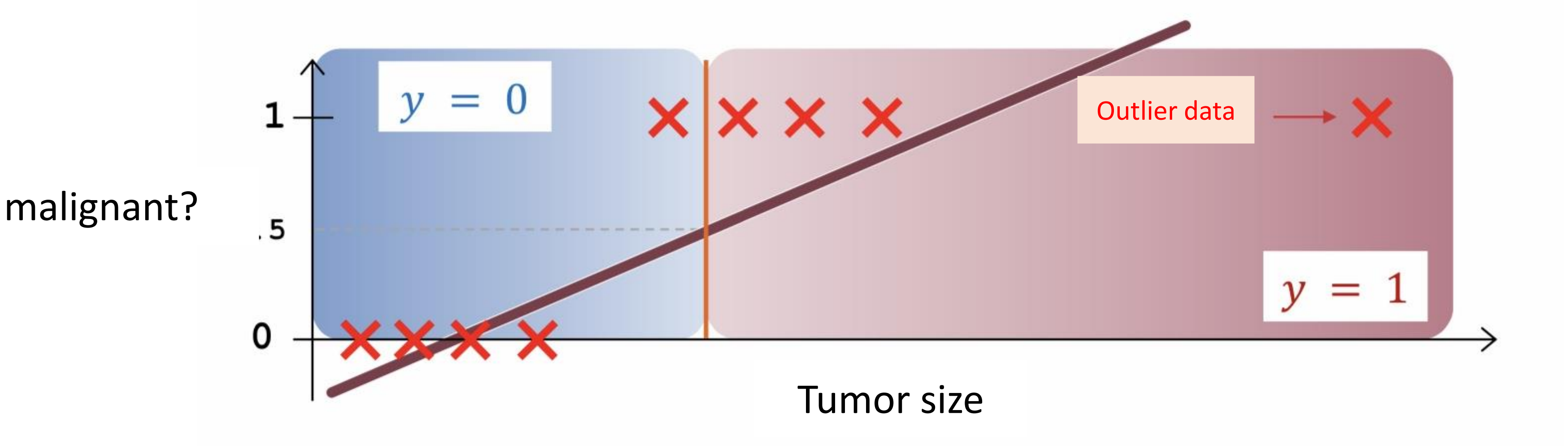
Classification

malignant?



- Placing a threshold on the classifier output:
  - If $h_\theta(x) \geq 0.5$, then y = 1
  - If $h_\theta(x) < 0.5$, then y = 0

Classification



Placing a threshold on the classifier output:

If $h_\theta(x) \geq 0.5$, then y = 1

If $h_\theta(x) < 0.5$, then y = 0

Classification

- In binary category we have:

$$y = 0 \text{ or } y = 1$$

- But in regression it is possible:

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

- Logistic regression (categorization)

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

Hypothesis representation in logistic regression

Hypothesis representation

target

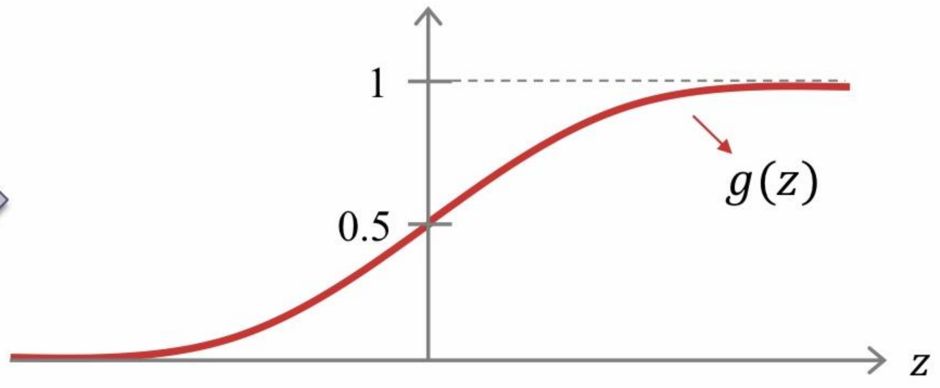
$$0 \le h_\theta(x) \le 1$$

Hypothesis

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

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

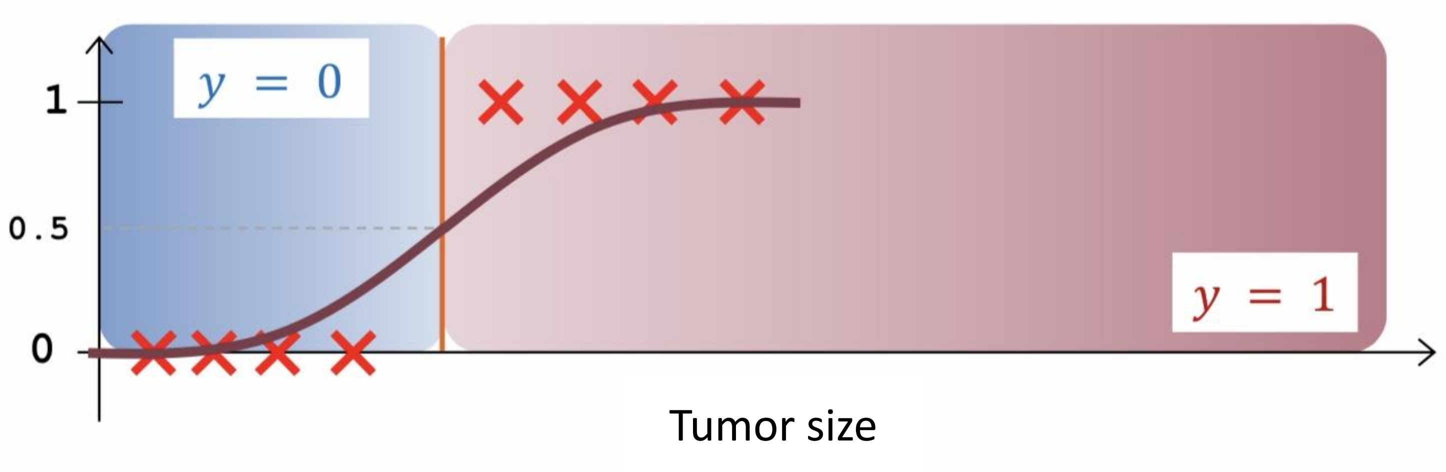
تابع سیگموید
(لجستیک)



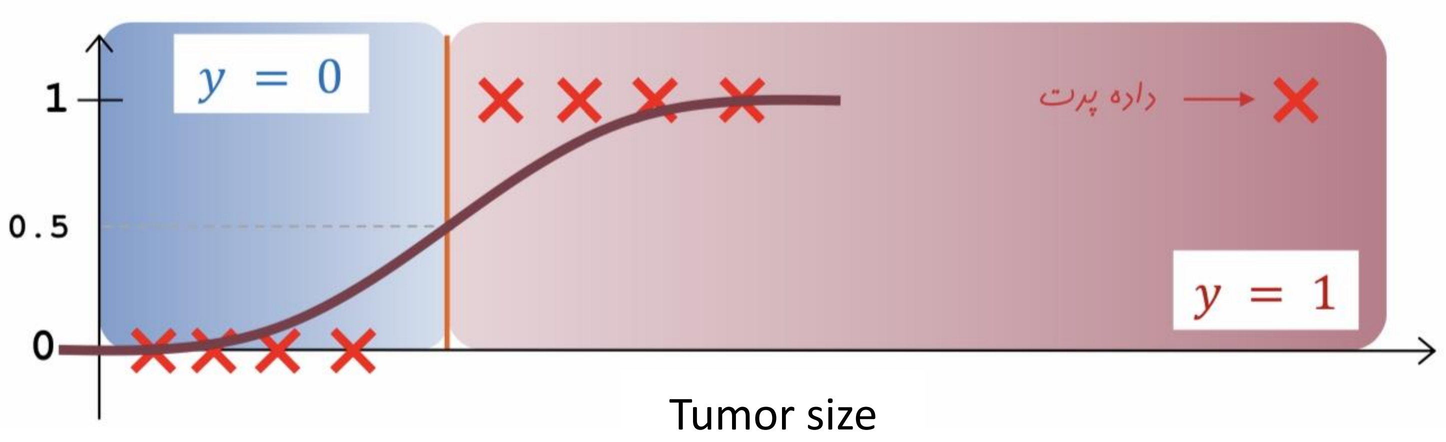
$$0 \le \boxed{g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}} \le 1$$

Logistic Regression and Classification

malignant?



malignant?



Hypothesis

- Interpretation of the output of the hypothesis:
- "Probability that input x belongs to category y = 1."
- Example: if we have:

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ tumorSize \end{bmatrix}, \qquad h_\theta(x) = 0.7$$

- In this case, there is a 70% chance that this tumor is malignant.

$$p(y = 1|x; \theta) = h_\theta(x)$$

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

Hypothesis

Probabilistic interpretation of the hypothesis

$$p(y = 1|x;\theta) = h_\theta(x)$$

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

$$p(y|x;\theta) = h_\theta(x)^y \big(1 - h_\theta(x)\big)^{1-y}$$

Likelihood function

$$L(\theta) = p(Y|X;\theta) = \prod_{i=1}^{m} p\left(y^{(i)}\middle|x^{(i)};\theta\right)$$

$$= \prod_{i=1}^{m} h_\theta\left(x^{(i)}\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}$$

Maximum likelihood estimation

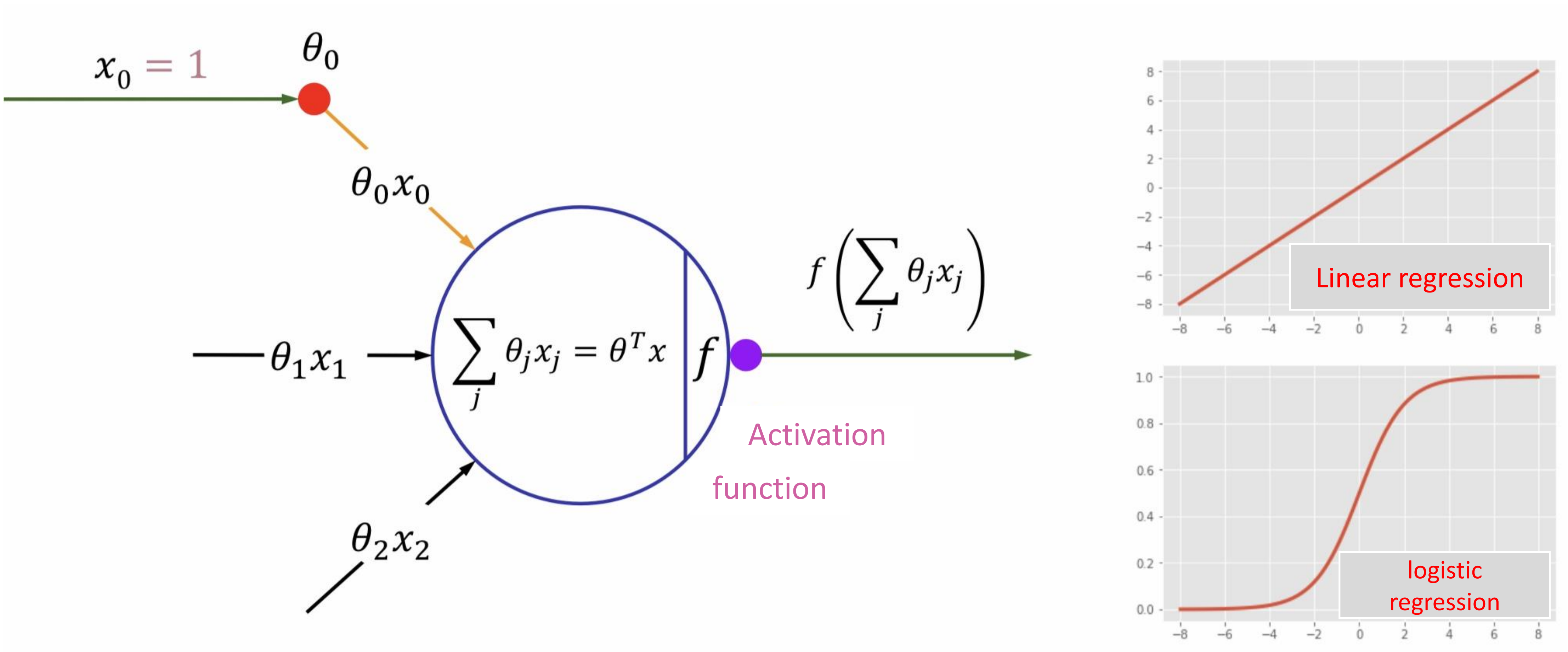
The logarithm of the likelihood function:

$$
\begin{aligned}
l(\theta) = \log L(\theta) &= \log \prod_{i=1}^{m} h_\theta\left(x^{(i)}\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}} \\
&= \sum_{i=1}^{m} \log\left(h_\theta\left(x^{(i)}\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}\right) \\
&= \sum_{i=1}^{m} y^{(i)} \log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)
\end{aligned}
$$

Cost function:

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

Linear regression and logistic regression



Decision Boundary

Decision Boundary

logistic regression:

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

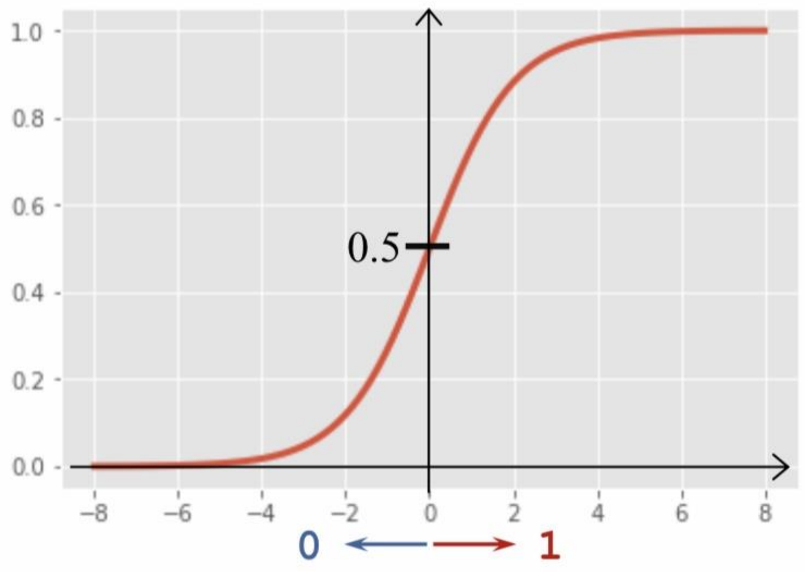
Placing a threshold on the bundle output:

$$y = 1:\ h_\theta(x) \geq 0.5 \Rightarrow \theta^T x \geq 0$$

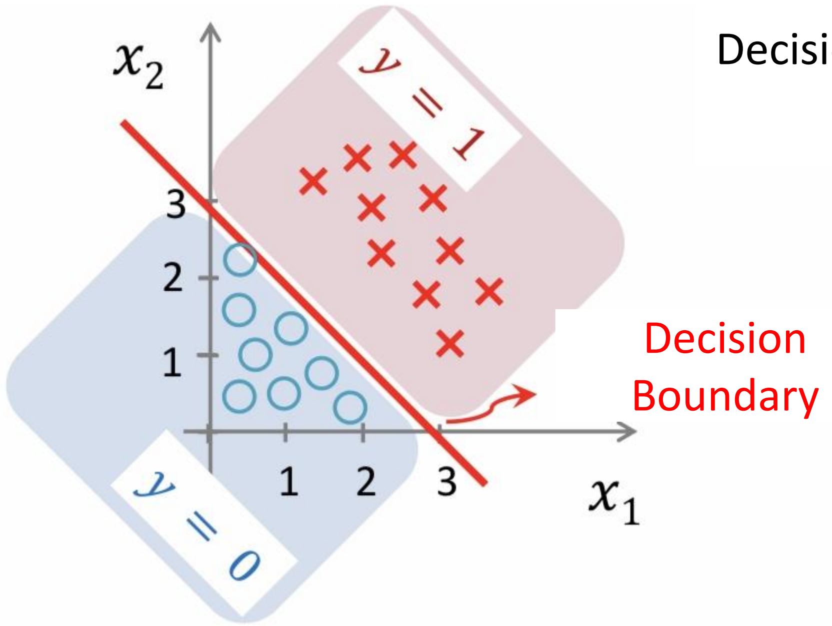
$$y = 0:\ h_\theta(x) < 0.5 \Rightarrow \theta^T x < 0$$

$$\boxed{\theta^T x = 0}$$

Decision boundary equation



Decision Boundary



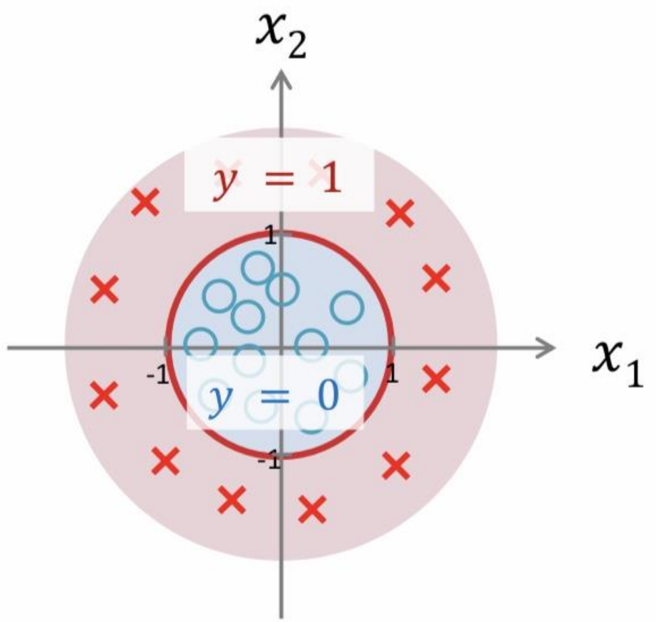
Decision Boundary:

$$h_\theta(x) = g(\underset{-3}{\theta_0} + \underset{1}{\theta_1} x_1 + \underset{1}{\theta_2} x_2)$$

The output of y is equal to 1, if $-3 + x_1 + x_2 \geq 0$

- $x_1 + x_2 \geq 3 \Rightarrow y = 1$
- $x_1 + x_2 < 3 \Rightarrow y = 0$

Non-linear Decision Boundary



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

-1 0 0 1 1

$$x_1^2 + x_2^2 \geq 1 \Rightarrow y = 1$$

$$x_1^2 + x_2^2 < 1 \Rightarrow y = 0$$

Cost Function

Logistic Regression

Training Set

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

Training Example

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

Hypothesis

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

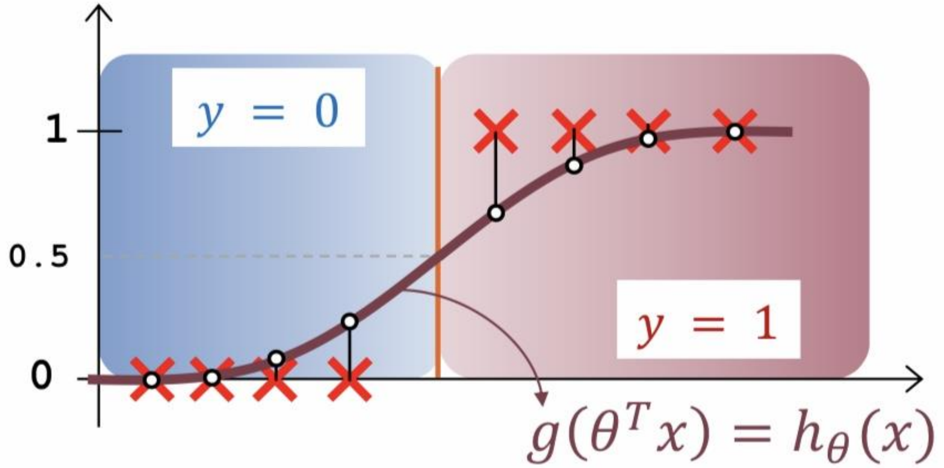
How to choose values of the theta parameters?

Logistic Regression

Cost function

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

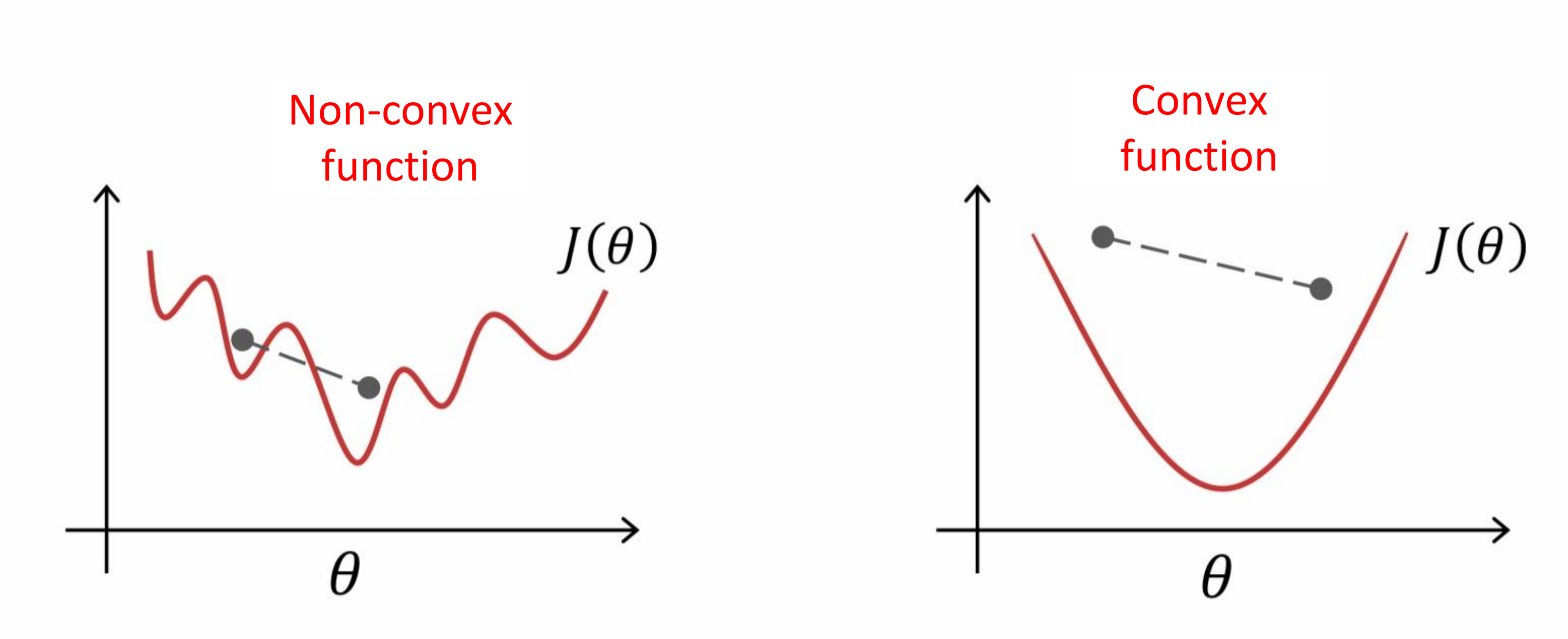
$$cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = \frac{1}{2}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$



Attention: Since $h_\theta(x^{(i)})$ is a non-linear function of parameters, the cost function will no longer be a convex function.

Cost Function

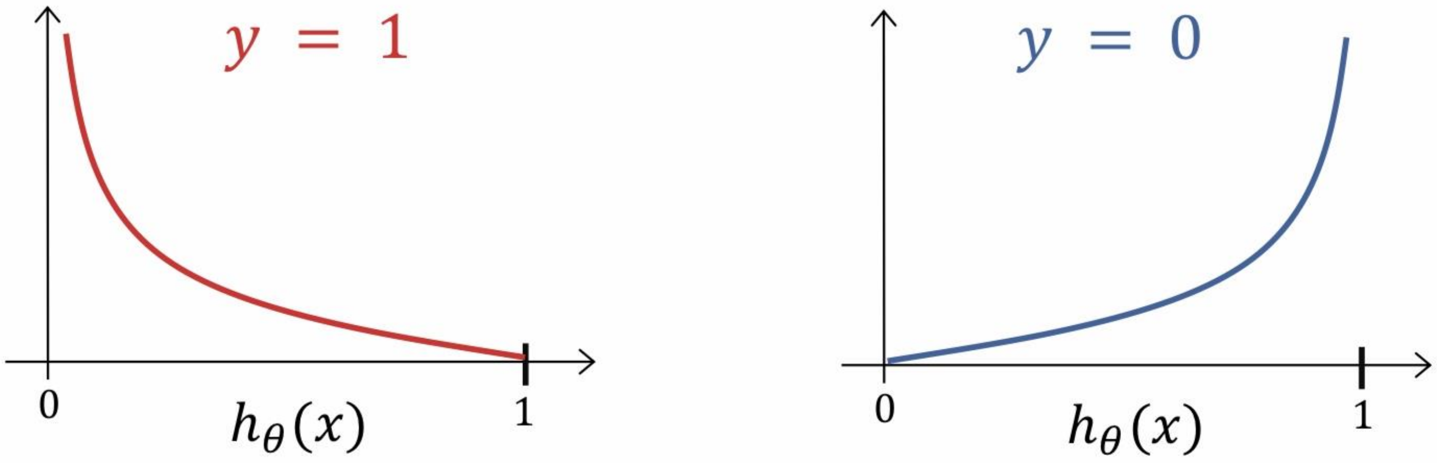
- Convex and non-convex functions



Cost Function in Logistic Regression

- Cost function

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

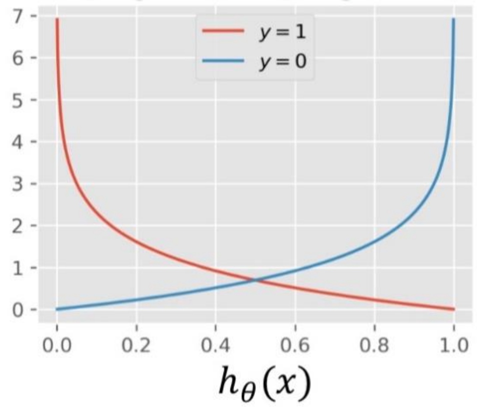


Cost Function in Logistic Regression

- Cost function simplification

$$J(\theta) = \sum_{i=1}^{m} cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)$$

$$cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) = -y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) - \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$



Cost Function in Logistic Regression

Cost function

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

Parameters'
Value specification

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

Prediction for
New input x

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

Cost Function in Logistic Regression

- Cost function

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

$$\nabla J(\theta) = X^T\left(h_\theta(X) - y\right) \qquad \nabla J(\theta) \in \mathbb{R}^{n+1}$$

$$H = X^T \operatorname{diag}\left(h_\theta(X)\left(1 - h_\theta(X)\right)\right) X \qquad H \in \mathbb{R}^{(n+1)\times(n+1)}$$

Note: the Hessian matrix is a positive definite matrix, so the cost function is a convex function.

Gradient Descent Algorithm

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

Gradient Descent Algorithm
(vector form)

repeat until convergence {

$$\theta := \theta - \alpha \underbrace{\nabla J(\theta)}$$

}

$$\nabla J(\theta) = X^T (h_\theta(X) - y)$$

Gradient Descent Algorithm

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

Gradient Descent Algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \underbrace{\frac{\partial}{\partial \theta_j} J(\theta)} \qquad (j = 0,1, \ldots, n)$$

}

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

Gradient Descent Algorithm

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

Gradient Descent Algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \qquad (j = 0,1,\ldots,n)$$

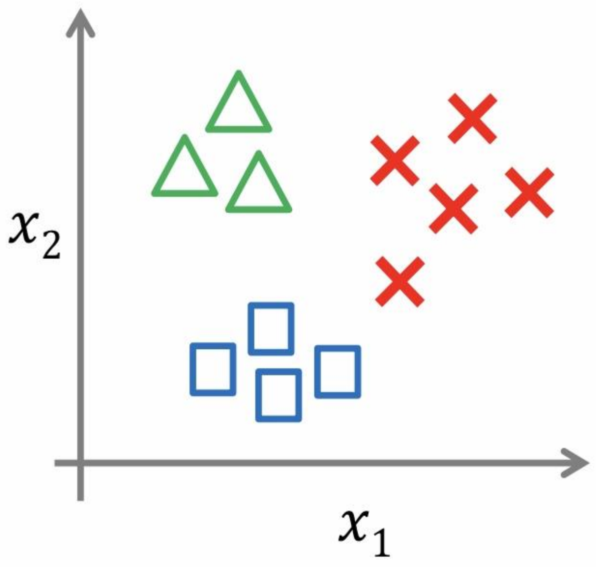
}

Note: This algorithm is just like the linear regression algorithm and the only difference is in the hypothesis function.

Classification with more than two classes

Classification with more than two classes

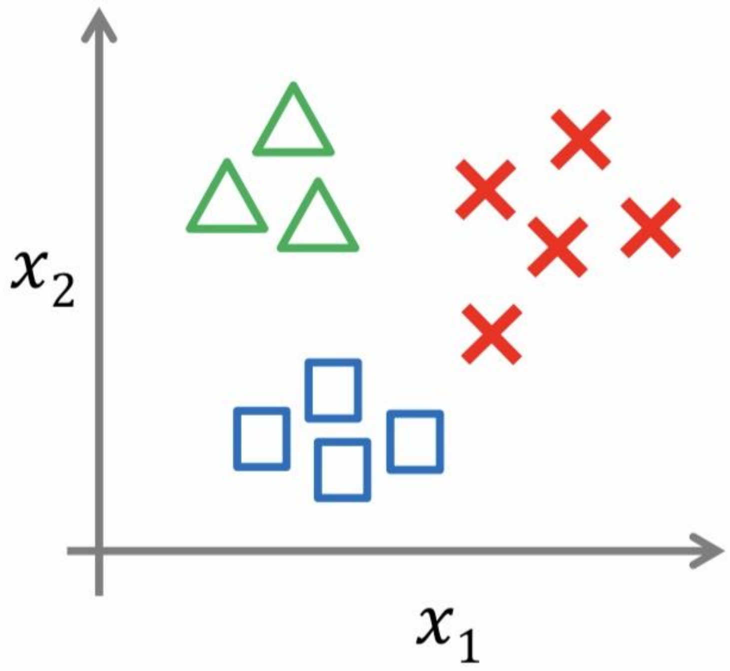
Email: work, family, entertainment
Medical charts: healthy, cold, flu
Weather: sunny, cloudy, rainy, snowy



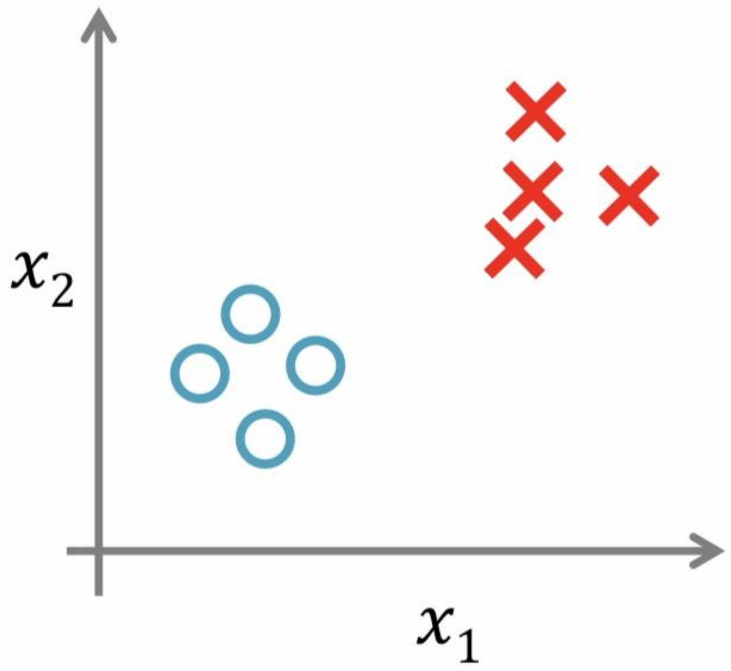
$$y \in \{1, 2, 3, \ldots, k\}$$

Classification with more than two classes

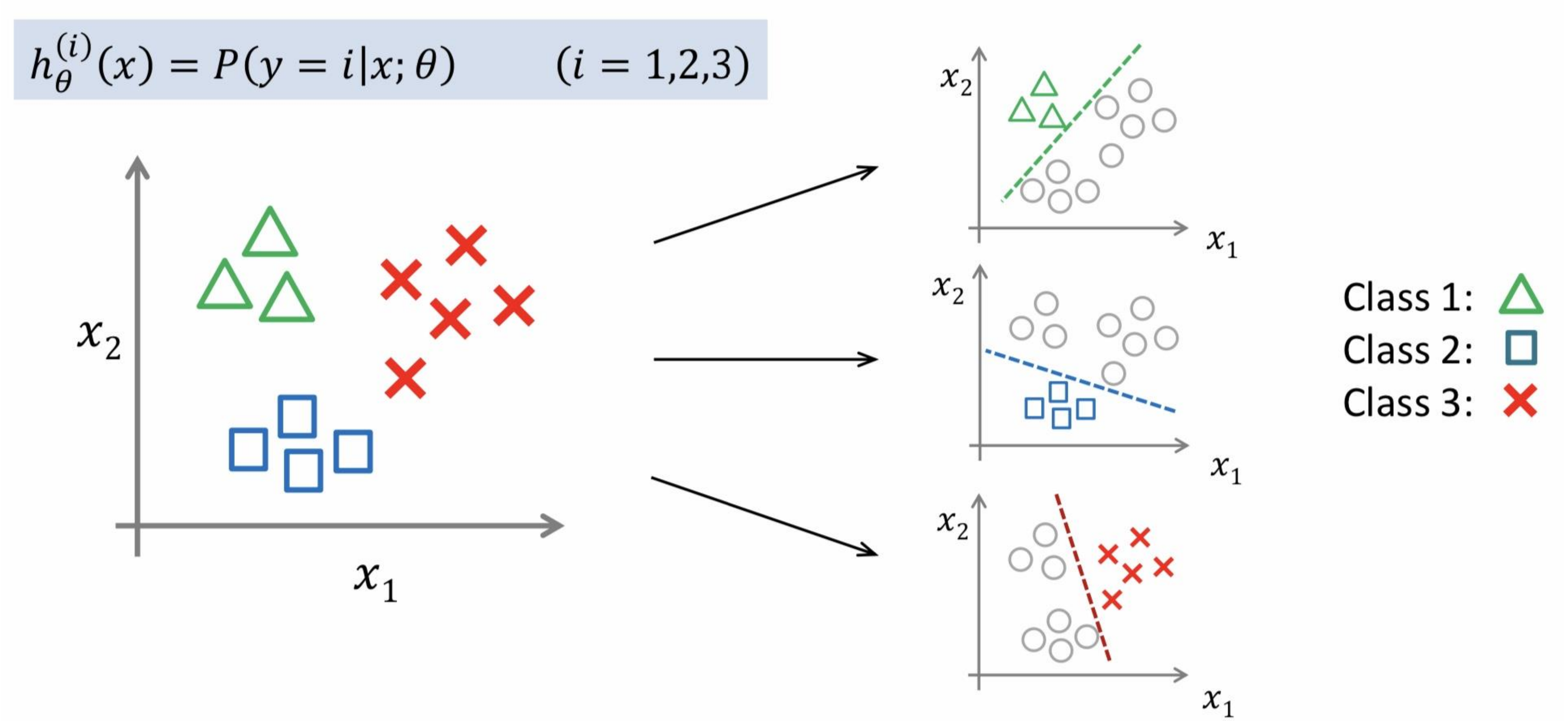
Multi-class classification



Binary classification



Classification with more than two classes: one to many



Classification with more than two classes: one to many

- One vs. All: For each category i, train the logistic regression classifier $h_\theta^{(i)}(x)$ to estimate the probability that input x belongs to category i.
- Prediction: In order to categorize the new input x, choose category i such that:

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

$$h_\theta^{(1)}(x) = 0.25$$

$$\boxed{h_\theta^{(2)}(x) = 0.70} \Rightarrow y = 2$$

$$h_\theta^{(3)}(x) = 0.45$$

Advanced optimization methods

Advanced optimization methods

Objective: To find the value of theta in order to minimize the cost function.

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

Assumption: we have a program that can calculate the following values by having theta values:

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

repeat until convergence {

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

}

Gradient Descent

Advanced optimization methods

- Assumption: we have a program that can calculate the following values by having theta values:

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

- Advanced optimization algorithms:
  - Conjugate gradient
  - BFGS
  - L-BFGS
- Advantages: these methods do not need to choose the learning rate and usually converge earlier than the gradient descent algorithm.

Advanced optimization methods

- Example

$$J(\theta) = (\theta_0 - 5)^2 + (\theta_1 - 5)^2 \Rightarrow \begin{aligned} \frac{\partial}{\partial \theta_0} J(\theta) &= 2(\theta_0 - 5) \\ \frac{\partial}{\partial \theta_1} J(\theta) &= 2(\theta_1 - 5) \end{aligned}$$

```
def J(theta):
    return (theta[0] - 5) ** 2 + (theta[1] - 5) ** 2
```

```
def grads(theta):
    return np.array([2 * (theta[0] - 5), 2 * (theta[1] - 5)])
```

Advanced optimization methods

- Example

$$J(\theta) = (\theta_0 - 5)^2 + (\theta_1 - 5)^2$$

$$\frac{\partial}{\partial \theta_0} J(\theta) = 2(\theta_0 - 5)$$

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

```
from scipy.optimize import minimize

minimize(J, x0=[0, 0], method='CG', jac=gra
```

```
     fun: 2.477476329894505e-18
     jac: array([1.71271335e-08, 1.71271335e-08])
 message: 'Optimization terminated successfully.'
    nfev: 20
     nit: 2
    njev: 5
  status: 0
 success: True
       x: array([5., 5.])
```

Advanced optimization methods

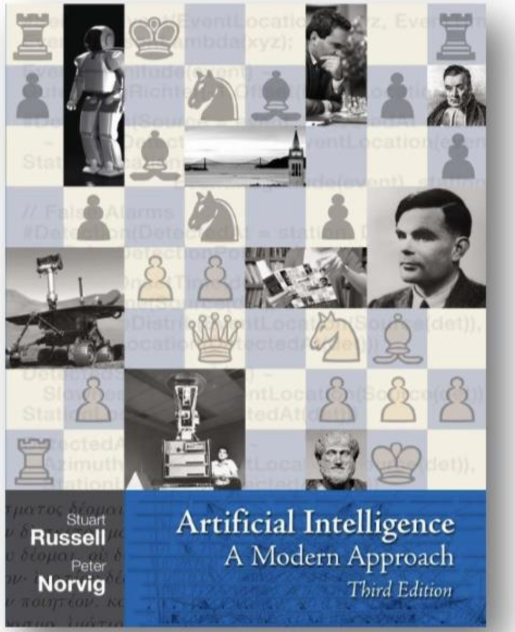
- Example

$$J(\theta) = (\theta_0 - 5)^2 + (\theta_1 - 5)^2 \quad \Rightarrow \quad \frac{\partial}{\partial \theta_0} J(\theta) = 2(\theta_0 - 5), \quad \frac{\partial}{\partial \theta_1} J(\theta) = 2(\theta_1 - 5)$$

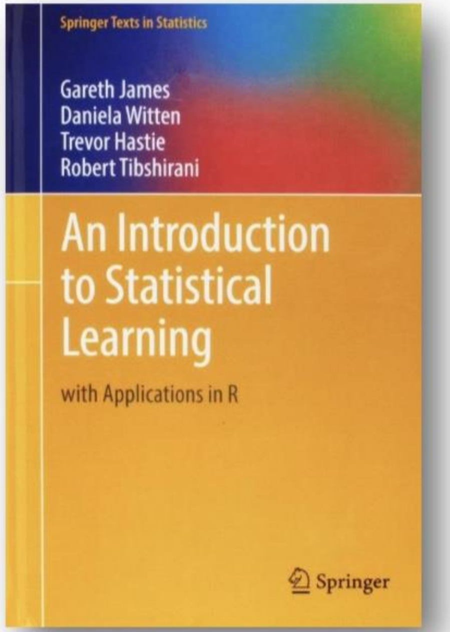
```python
from scipy.optimize import minimize

minimize(J, x0=[0, 0], method='BFGS', jac=
```

```
     fun: 3.5538794606501983e-16
hess_inv: array([[ 0.75, -0.25],
       [-0.25,  0.75]])
     jac: array([-1.17592194e-08, -1.17592194e-08])
 message: 'Optimization terminated successfully.'
    nfev: 16
     nit: 3
    njev: 4
  status: 0
 success: True
       x: array([4.99999999, 4.99999999])
```

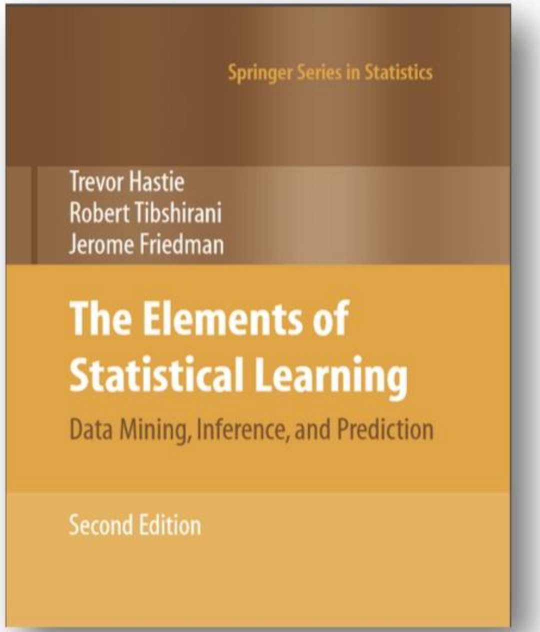
Further study



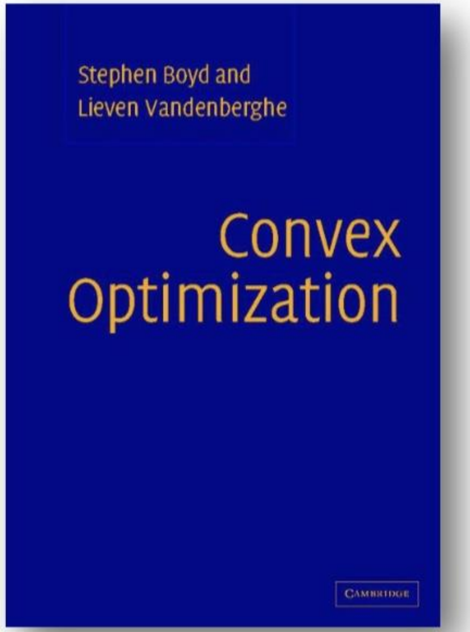
Pages 725-727



Pages 130-137



Pages 119-128



Convex optimization