Logistic regression (classification)

Duration: 3 hrs

- Classification
- Hypothesis representation
- Decision boundary
- Cost function
- Multi-class classification: One-vs-all

Source: Machine Learning, Andrew Ng, coursera.org

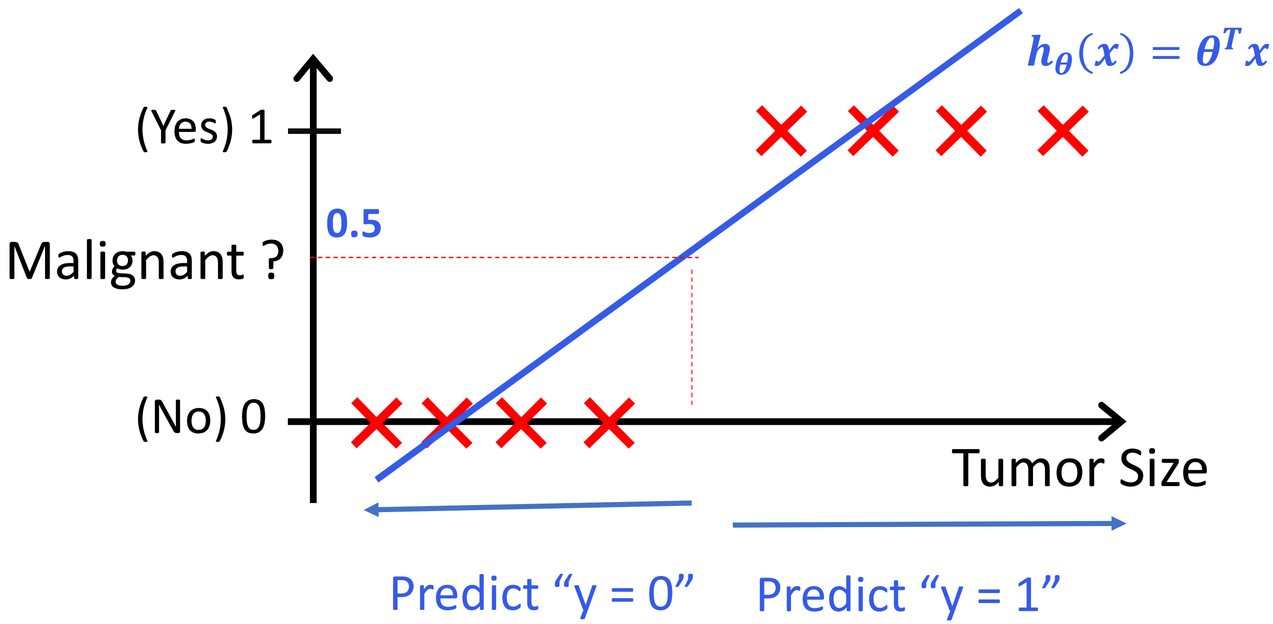
Classification

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)

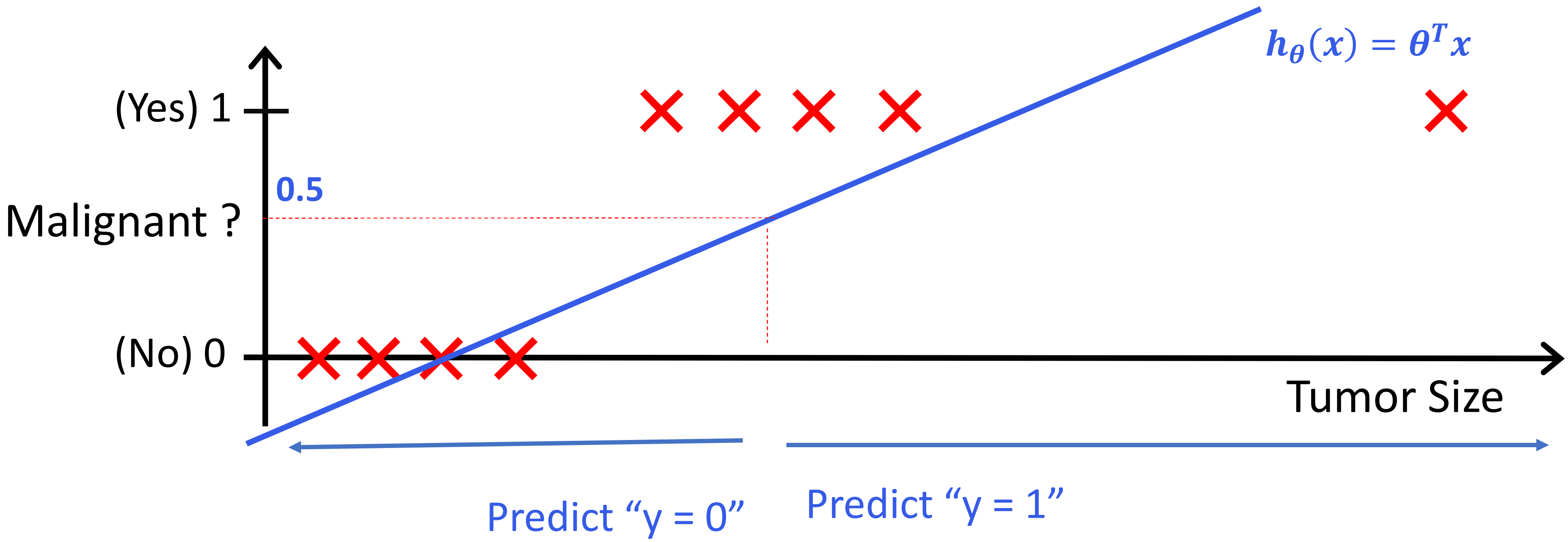
$y \in \{0, 1, 2, 3\}$



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"



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

Logistic regression (classification)

- Classification
- Hypothesis representation
- Decision boundary
- Cost function
- Multi-class classification: One-vs-all

Source: Machine Learning, Andrew Ng, coursera.org

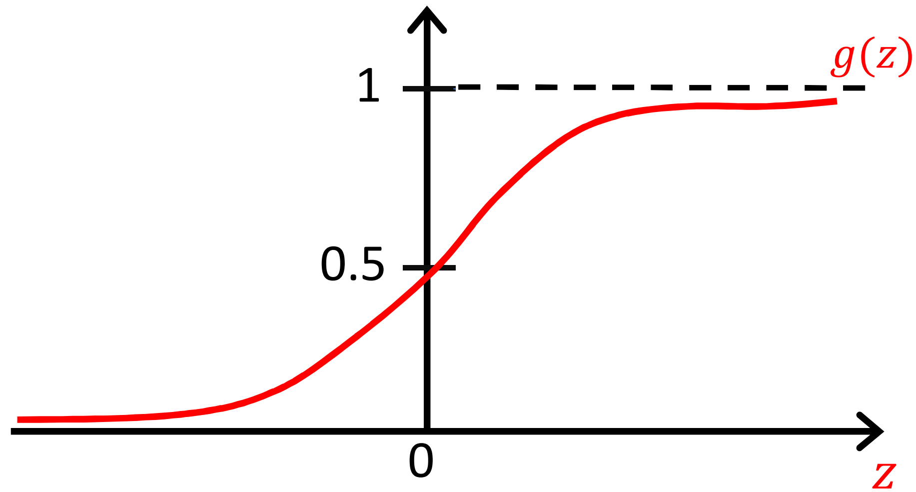
Logistic Regression Model

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

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

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

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



Sigmoid function

Logistic function

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

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

"probability that y = 1, given x, parameterized by θ"

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

Logistic regression (classification)

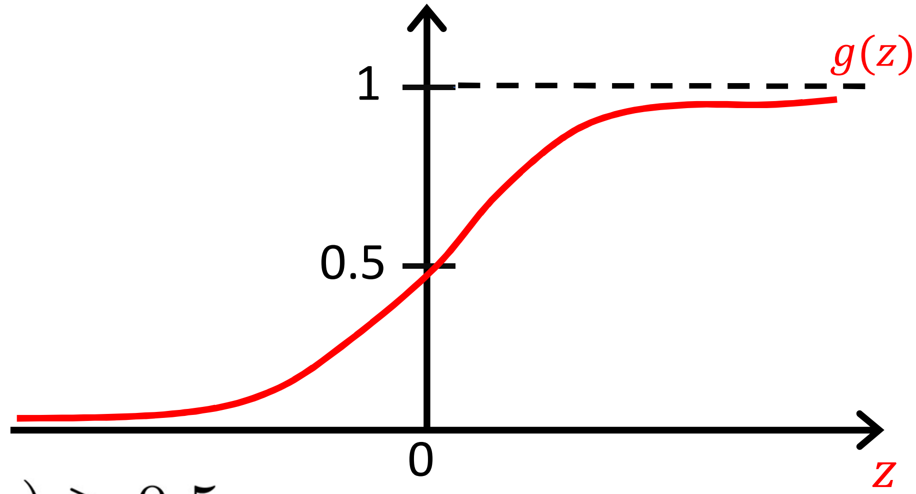
- Classification
- Hypothesis representation
- Decision boundary
- Cost function
- Multi-class classification: One-vs-all

Source: Machine Learning, Andrew Ng, coursera.org

Logistic regression

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

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



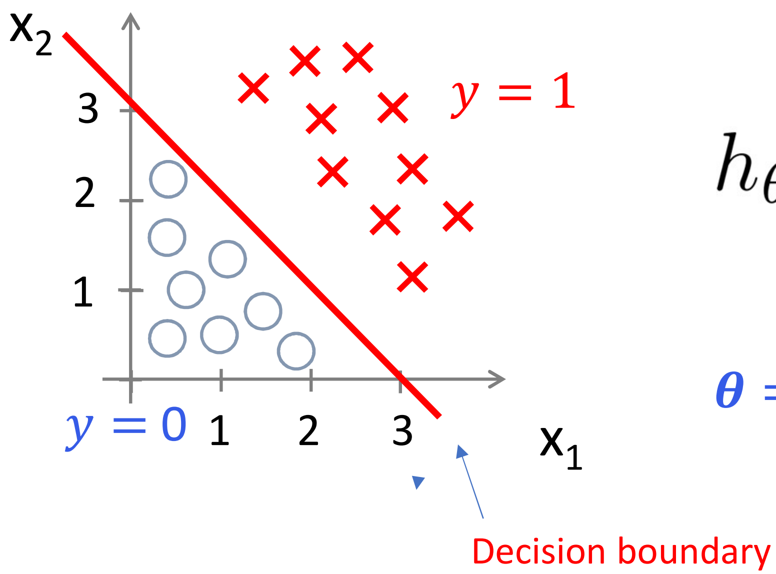
Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

$when\ z \geq 0\ or\ \theta^T x \geq 0$

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

$when\ z < 0\ or\ \theta^T x < 0$

Decision Boundary



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

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

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

$x_1 + x_1 \geq 3$

$h_\theta(x) = 0.5$

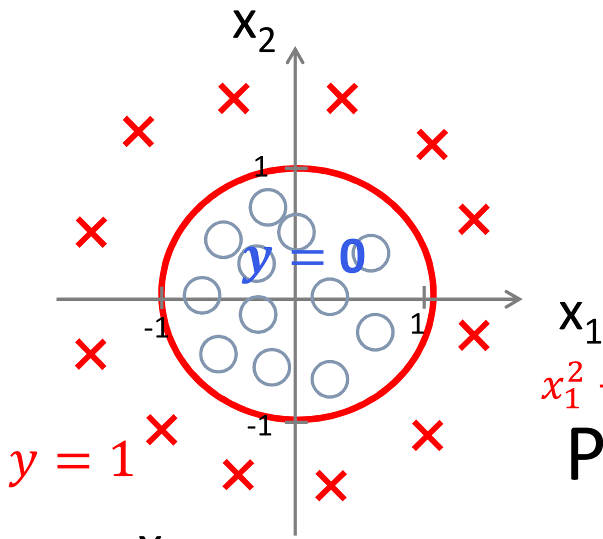
$x_1 + x_1 = 3$

Decision boundary

$x_1 + x_1 < 3$

$y = 0$

Non-linear decision boundaries



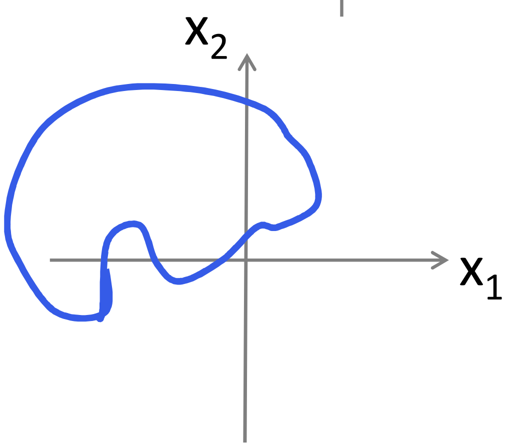
$x_1^2 + x_1^2 = 1$

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

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

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

$x_1^2 + x_1^2 \geq 1$



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

Logistic regression (classification)

- Classification
- Hypothesis representation
- Decision boundary
- Cost function

Source: Machine Learning, Andrew Ng, coursera.org

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

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

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

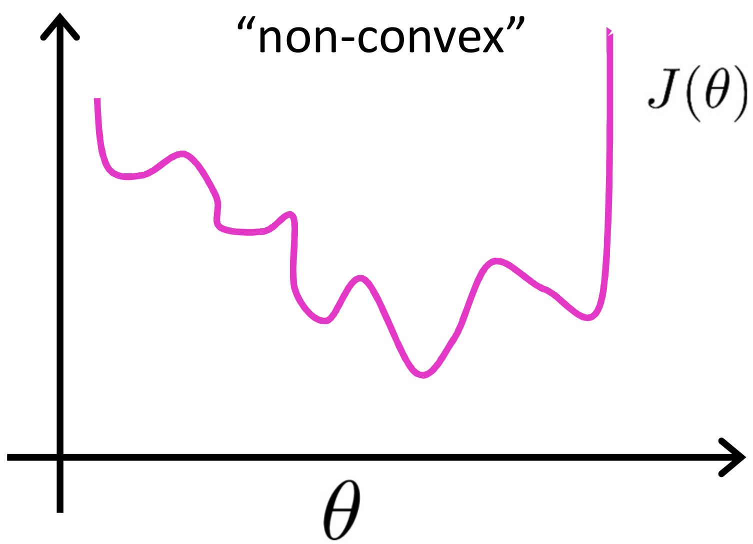
How to choose parameters θ ?

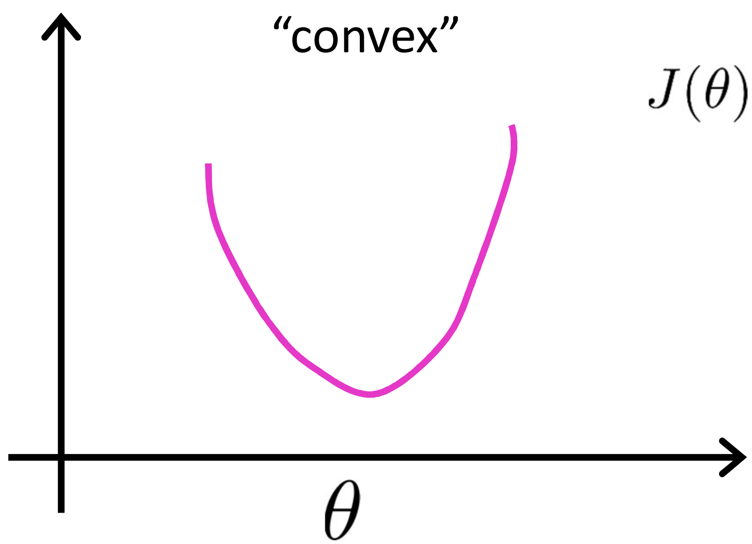
Cost function

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

Logistic regression: $cost(h_\theta(x), y) = \frac{1}{2}(h_\theta(x) - y)^2$

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

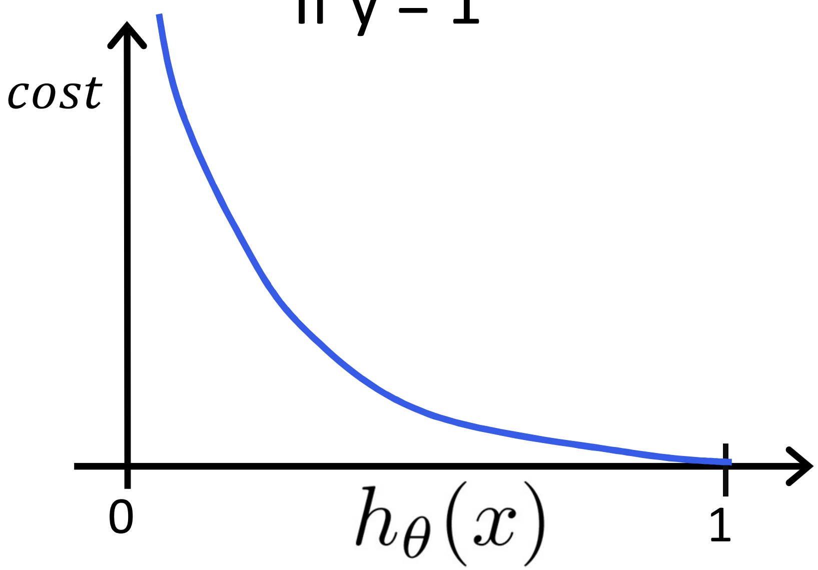




Logistic regression cost function

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

If y = 1



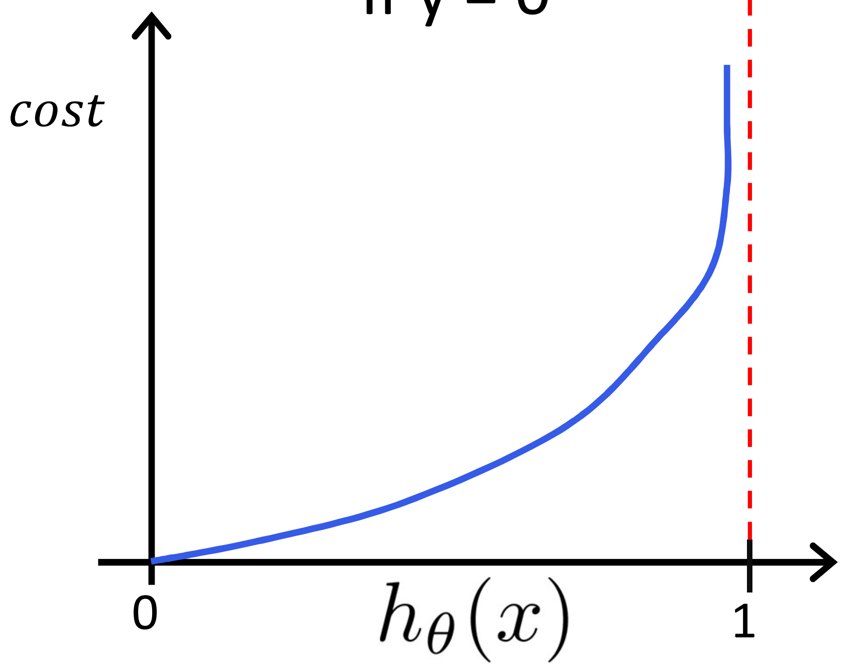
$\mathrm{Cost} = 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

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

Logistic regression cost function

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

If y = 0



$cost$ = 0 if $y = 0, h_\theta(x) = 0$

But as $h_\theta(x) \rightarrow 1$

$cost \rightarrow \infty$

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

Logistic regression (classification)

- Classification
- Hypothesis representation
- Decision boundary
- Cost function
    - Simplified cost function and gradient descent
- Multi-class classification: One-vs-all

Source: Machine Learning, Andrew Ng, coursera.org

Logistic regression cost function

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

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

Note: $y = 0$ or 1 always

$$cost(h_\theta(x), y) = -y.\log\big(h_\theta(x)\big) - (1-y).log(1 - h_\theta(x))$$

If y = 1: $cost(h_\theta(x), y) = -\log\big(h_\theta(x)\big)$

If y = 0: $cost(h_\theta(x), y) = -log(1 - h_\theta(x))$

Logistic regression cost function

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

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

To fit parameters θ:

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

To make a prediction given new x:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ $\qquad p(y = 1|x;\theta)$

Gradient Descent

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

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

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

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

Gradient Descent

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

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 θ_j)

$\}$

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

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

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

Algorithm looks identical to linear regression!

Logistic regression (classification)

- Classification
- Hypothesis representation
- Decision boundary
- Cost function
  - Simplified cost function and gradient descent
  - Advanced optimization
- Multi-class classification: One-vs-all

Source: Machine Learning, Andrew Ng, coursera.org

Optimization algorithm

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

Given θ, we have code that can compute

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

Gradient descent:

Repeat $\{$

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

$\}$

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

Advantages:

- No need to manually pick α
- Often faster than gradient descent.

Disadvantages:

- More complex

Example:

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

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

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

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

```
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 = 
```

$$\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

```
function [jVal, gradient] = costFunction(theta)

      jVal = [code to compute J(θ)];

      gradient(1) = [code to compute ∂/∂θ_0 J(θ)];

      gradient(2) = [code to compute ∂/∂θ_1 J(θ)];
         ⋮
      gradient(n+1) = [code to compute ∂/∂θ_n J(θ)   ];
```

$$\texttt{jVal = [} \text{code to compute } J(\theta)\texttt{];}$$

$$\texttt{gradient(1) = [} \text{code to compute } \frac{\partial}{\partial\theta_0} J(\theta)\texttt{];}$$

$$\texttt{gradient(2) = [} \text{code to compute } \frac{\partial}{\partial\theta_1} J(\theta)\texttt{];}$$

$$\vdots$$

$$\texttt{gradient(n+1) = [} \text{code to compute } \frac{\partial}{\partial\theta_n} J(\theta)\texttt{];}$$

Logistic regression (classification)

- Classification
- Hypothesis representation
- Decision boundary
- Cost function
- Multi-class classification: One-vs-all

Source: Machine Learning, Andrew Ng, coursera.org

Multiclass classification

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

$y = 1$ $\quad y = 2$ $\quad y = 3$ $\quad y = 4$

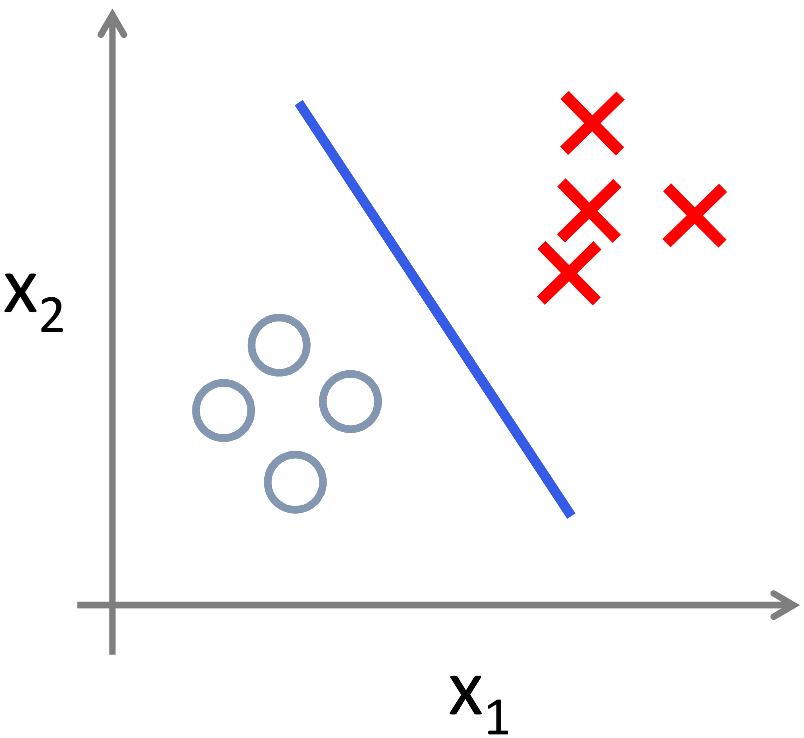
Medical diagrams: Not ill, Cold, Flu

$y = 1$ $\quad y = 2$ $\quad y = 3$

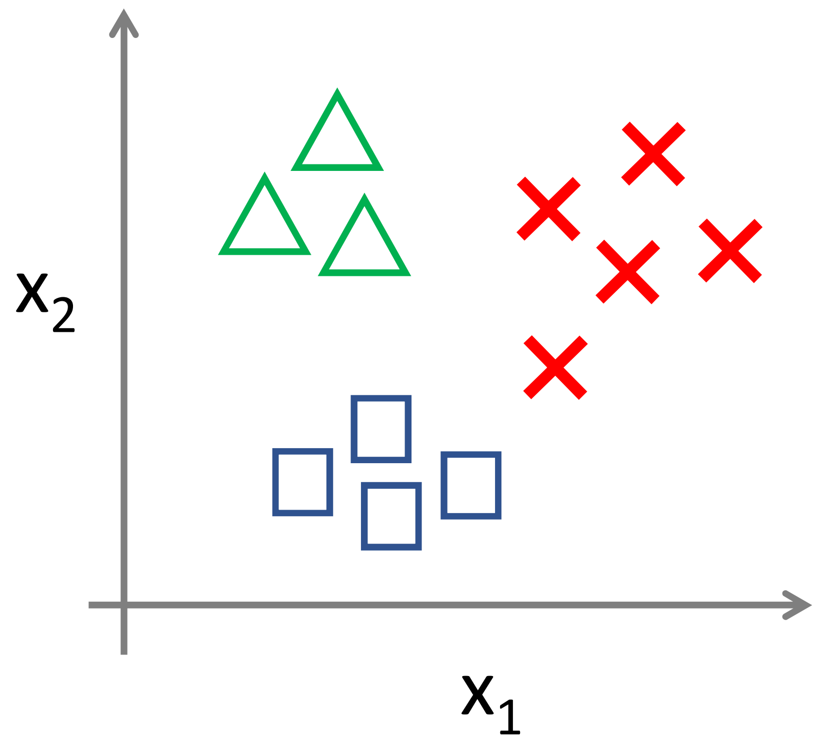
Weather: Sunny, Cloudy, Rain, Snow

$y = 1$ $\quad y = 2$ $\quad y = 3$ $\quad y = 4$

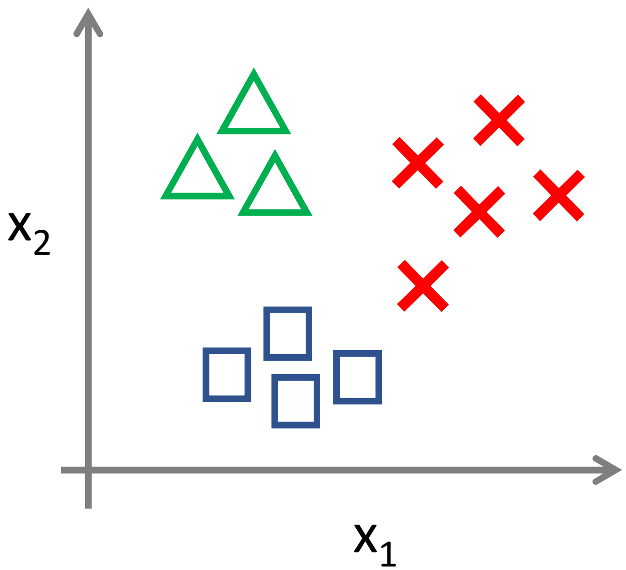
Binary classification:



Multi-class classification:



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

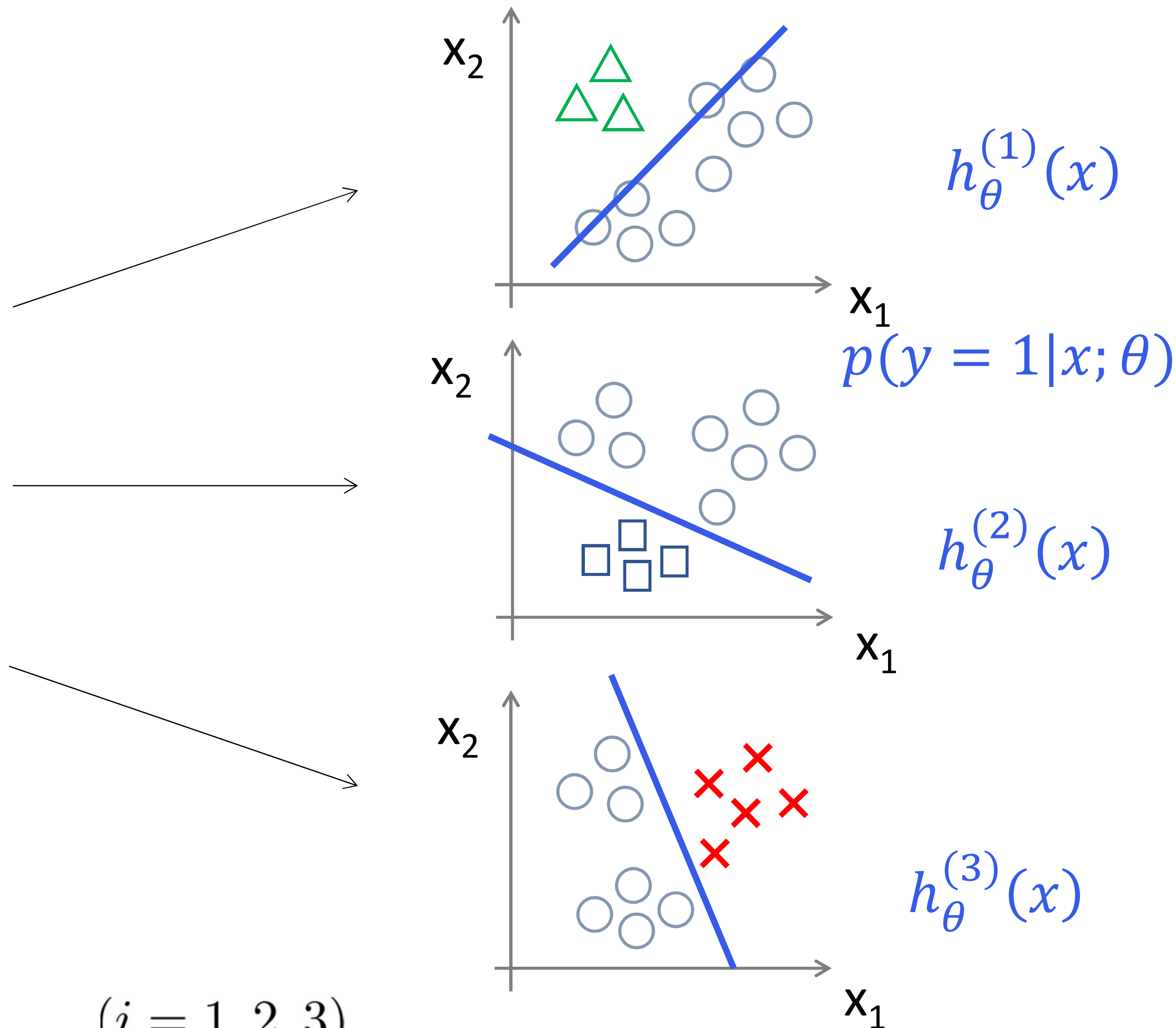


Class 1: △

Class 2: □

Class 3: ✕

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



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

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