Logistic Regression

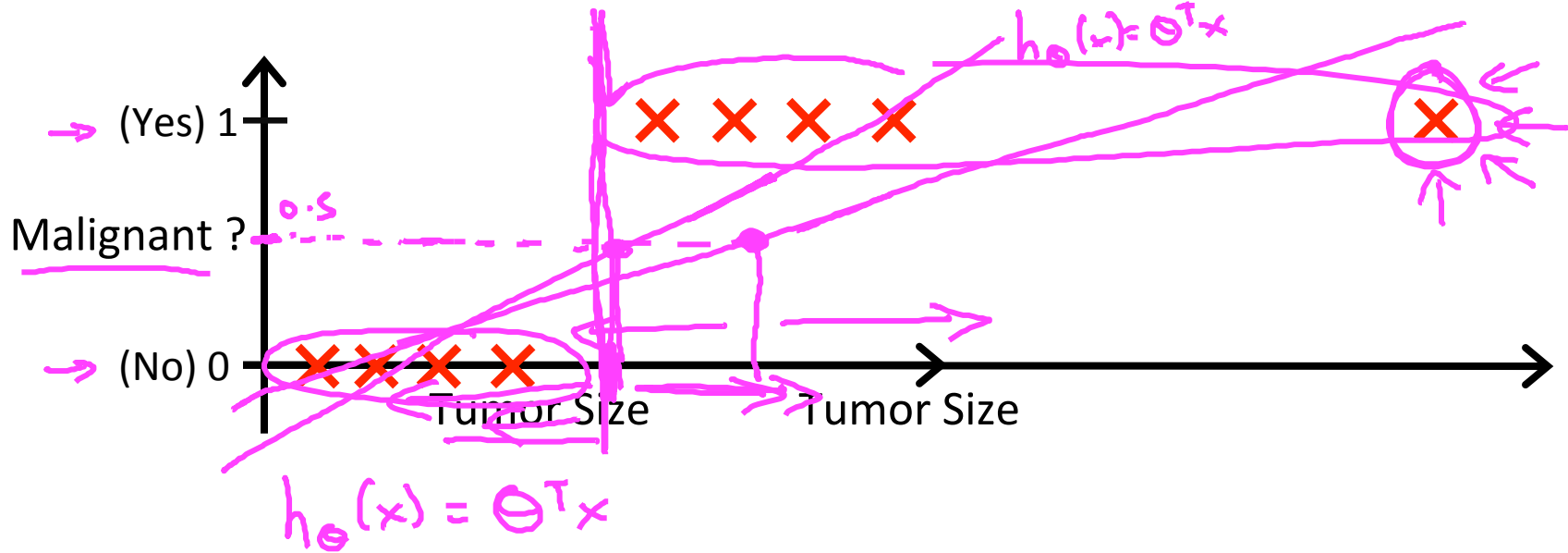## Classification

Machine Learning

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"

Classification: $y = 0$ or 1

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

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

Classification

Logistic Regression

Hypothesis Representation

Machine Learning

Logistic Regression Model

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

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

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

$\theta^T x$

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

Sigmoid function

Logistic function

$g(z)$

Parameters θ.

Interpretation of Hypothesis Output

$h_\theta(x)$

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

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

$h_\theta(x) = 0.7$

$y = 1$

Tell patient that 70% chance of tumor being malignant

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

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

$y = 0 \text{ or } 1$

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

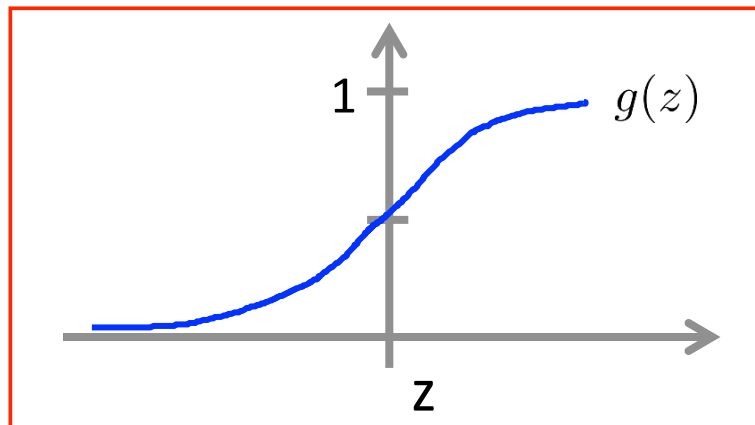$P(y = 0 | x; \theta) = 1 - P(y = 1 | x; \theta)$

Logistic Regression

Decision boundary

Machine Learning

Logistic regression

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

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

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

$\theta^T x \geq 0$

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

$\theta^T x < 0$

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

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

$g(z) < 0.5$
when $z < 0$

Decision Boundary

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

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

$$-3 \qquad 1 \qquad 1$$

Decision boundary

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

$\to x_1 + x_2 \geq 3$

x_1, x_2

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

$\boxed{x_1 + x_2 = 3}$

$x_1 + x_2 < 3$
$\to y = 0$

Andrew Ng

Non-linear decision boundaries

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

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

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

$$x_1^2 + x_2^2 = 1$$

$$x_1^2 + x_2^2 \geq 1$$

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

Andrew Ng

Logistic Regression

Cost function

Machine Learning

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

m examples

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \mathbb{R}^{n+1} \qquad x_0 = 1, y \in \{0, 1\}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$
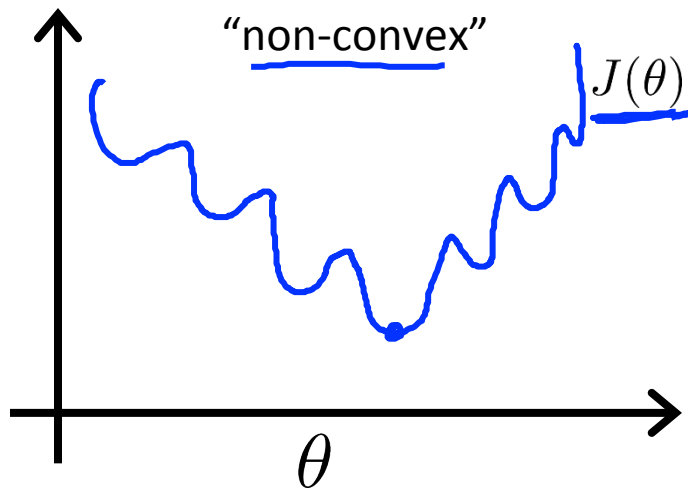
How to choose parameters θ ?

Cost function

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

Logistic

$\to \text{cost}(h_\theta(x^{(i)}), y)$

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

$\frac{1}{1 + e^{-\theta^T x}}$

"non-convex" $J(\theta)$

θ

"convex" $J(\theta)$

θ

Andrew Ng

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 = 1

$h_\theta(x)$

0   1

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

$\rightarrow$ 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

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

If y = 0

$-\log(1-z)$

Logistic Regression

Simplified cost function and gradient descent

Machine Learning

Logistic regression cost function

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

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

Note: $y = 0$ or 1 always

$\rightarrow \text{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y) \log(1 - h_\theta(x))$

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

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

Andrew Ng

Logistic regression cost function

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

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

To fit parameters θ :

$$\min_\theta J(\theta) \quad \text{Get } \theta$$

To make a prediction given new x:

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

$p(y = 1 \mid x; \theta)$

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 θ_j)

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

Gradient Descent

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

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

Repeat {

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

(simultaneously update all θ_j)

}

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \qquad \text{for } i = 0 \text{ to } n$$

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

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

Algorithm looks identical to linear regression!

Logistic Regression
Advanced optimization

Machine Learning

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

$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$ $\theta_1 = 5, \theta_2 = 5.$

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

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

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

```
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);
```

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

$$\mathbf{theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \begin{array}{l} \text{theta(1)} \leftarrow \\ \\ \text{theta(2)} \\ \\ \\ \text{theta(n+1)} \end{array}$$

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

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

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

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

$\vdots$

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

Logistic Regression

Multi-class classification: One-vs-all

Machine Learning

Multiclass classification

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

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

Medical diagrams: Not ill, Cold, Flu

$$y=1 \qquad 2 \qquad 3$$

Weather: Sunny, Cloudy, Rain, Snow

$$y=1 \qquad 2 \qquad 3 \qquad 4 \leftarrow$$

$$0 \qquad 1 \qquad 2 \qquad 3$$

Binary classification:

Multi-class classification:

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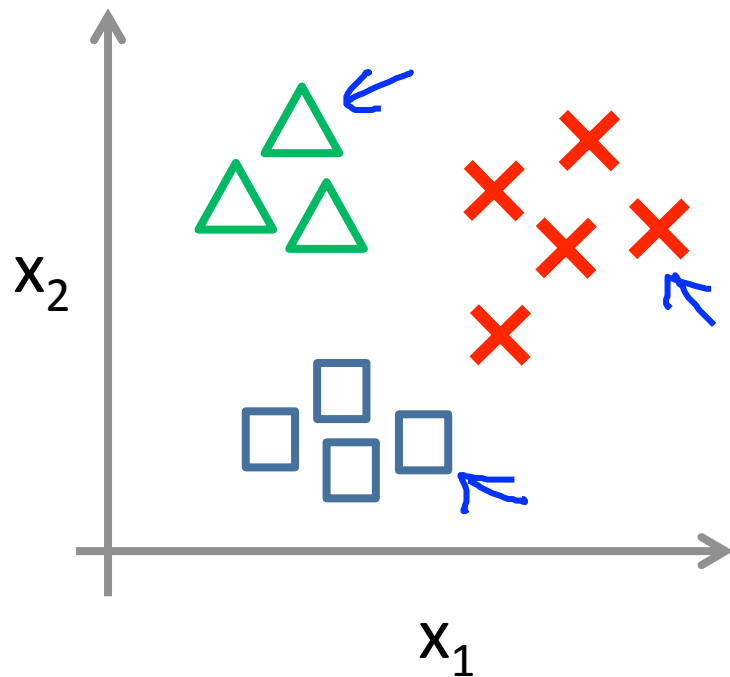
One-vs-all (one-vs-rest):

Class 1: △ ←
Class 2: □ ←
Class 3: ✕ ←

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

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

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

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

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

Andrew Ng

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

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