

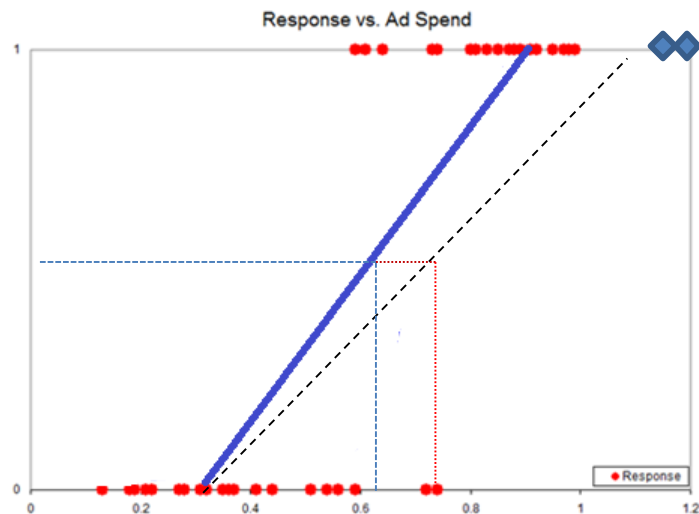
Logistic Regression

Data Science Dojo

Classification Types

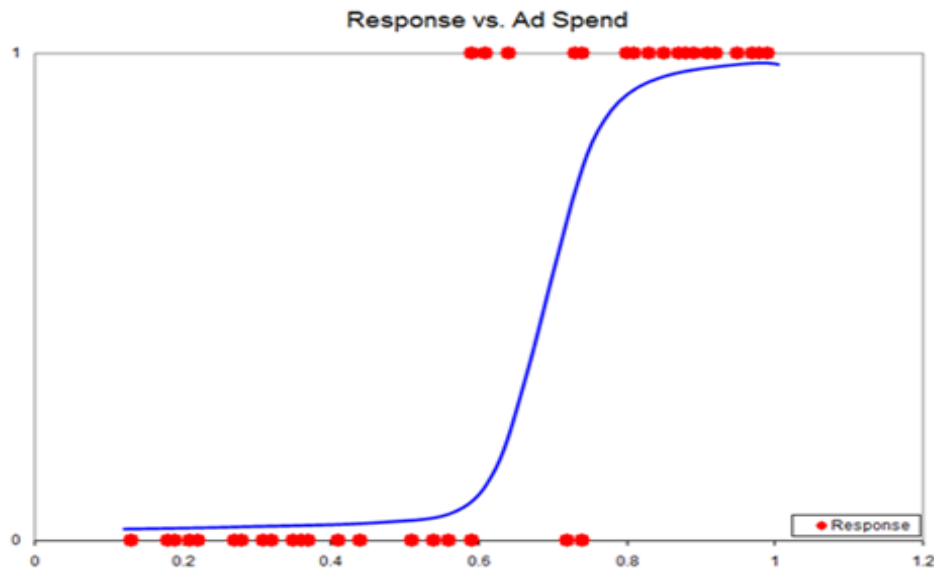
- Two-class
 - $y \in \{0,1\}$
 - 0: "negative" class
 - 1: "positive" class
 - Yes/No; Benign/Malignant; Click/No click
- Multi-class
 - $y \in \{0,1,2,...,n\}$
 - Grades; Colors; Item Categories

Two Class via Linear Regression



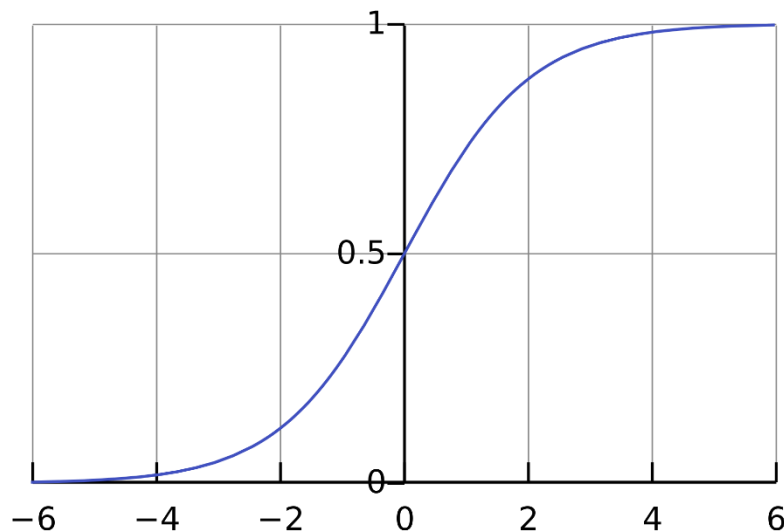
- Goal: Assign objects to $y = 0$ or $y = 1$
- Decision rule:
 - $h_{\theta}(x) \geq 0.5; y = 1$
 - $h_{\theta}(x) < 0.5; y = 0$
- Issues:
 - Hard to fit properly
 - $h_{\theta}(x)$ can be > 1 or < 0
 - Can't extract probabilities

Logistic Regression



- Use "Logistic Function"
- Shape better suited to two-class problem

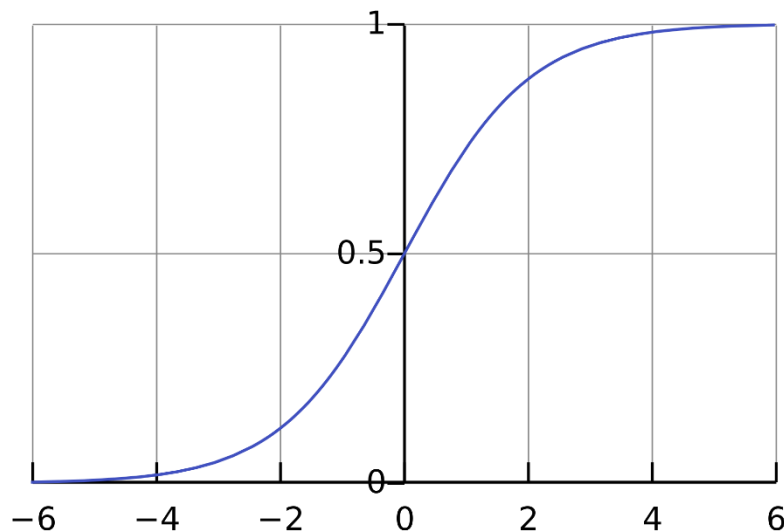
Logistic Function



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

- Class Assignment:
 - $y = 1$ if $h_{\theta}(x) \geq 0.5$
 - $y = 0$ if $h_{\theta}(x) < 0.5$
- Asymptotic
 - $h_{\theta}(x) \rightarrow 1$ as $x \rightarrow \infty$
 - $h_{\theta}(x) \rightarrow 0$ as $x \rightarrow -\infty$

Logistic Function



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

- Probabilistic interpretation
 - $P_{\theta}(y = 1|x) = h_{\theta}(x)$
 - $P_{\theta}(y = 0|x) = 1 - h_{\theta}(x)$
 - Always between 0 and 1!

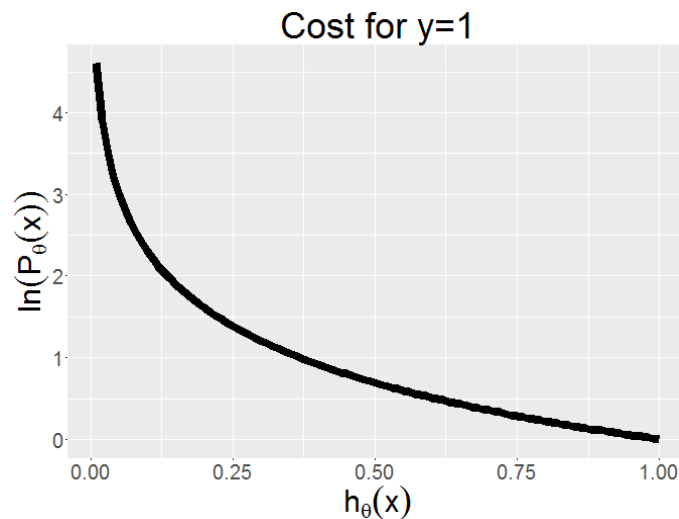
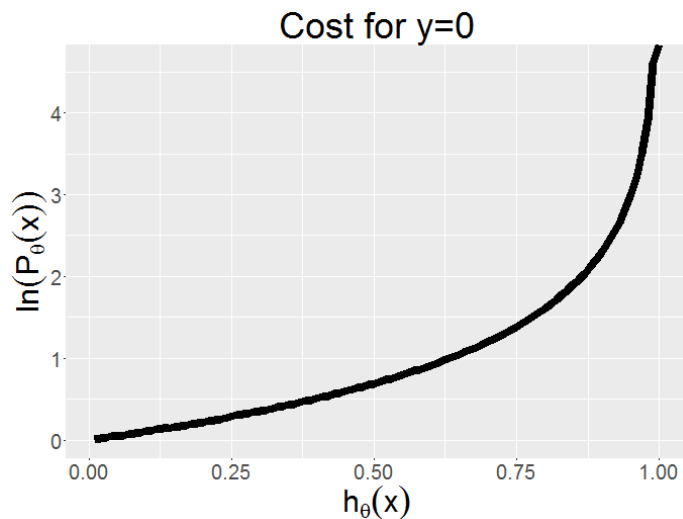
Reminder: Gradient Descent

$$\theta_{j+1} = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- α is the “learning rate”
- Each time, the algorithm takes a step in the direction of the steepest downward slope, so $J(\theta)$ decreases
- α determines how quickly or slowly the algorithm will converge to a solution

Cost Function: Log-Likelihood

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m -\ln(P_{\theta}(y = y_i | x^{(i)}))$$



Cost Function: Log-Likelihood

Need to find $\frac{\partial}{\partial \theta_j} J(\theta)$ for log-likelihood

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m -\ln \left(P_{\theta}(y = y_i | x^{(i)}) \right)$$

Since y is only 1 or 0, rewrite this as:

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

Quick Recap

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

Cost Function:
$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m y_i \ln(h_{\theta}(x^{(i)})) + (1 - y_i) \ln(1 - h_{\theta}(x^{(i)})) \right]$$

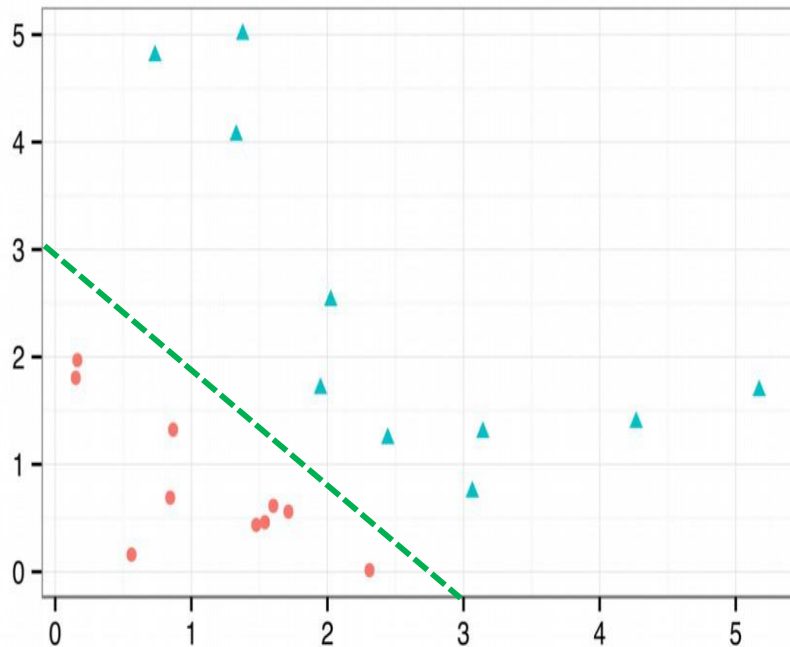
Parameters:
$$\theta^T = [\theta_1, \dots, \theta_n]$$

Decision Boundary

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

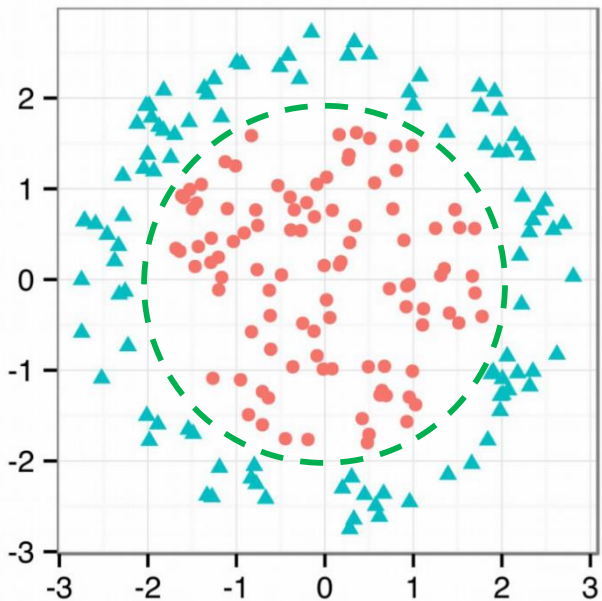
- Consider the exponent $\theta^T x$
 - Assume threshold of 0.5
 - $0.5 = \frac{1}{1 + e^{-\theta^T x}} \rightarrow e^{-\theta^T x} = 1 \rightarrow \theta^T x = 0$
 - $D(\theta, x) = \theta^T x$ is the "decision boundary"
 - Visualize class separation

Decision Boundary: Linear



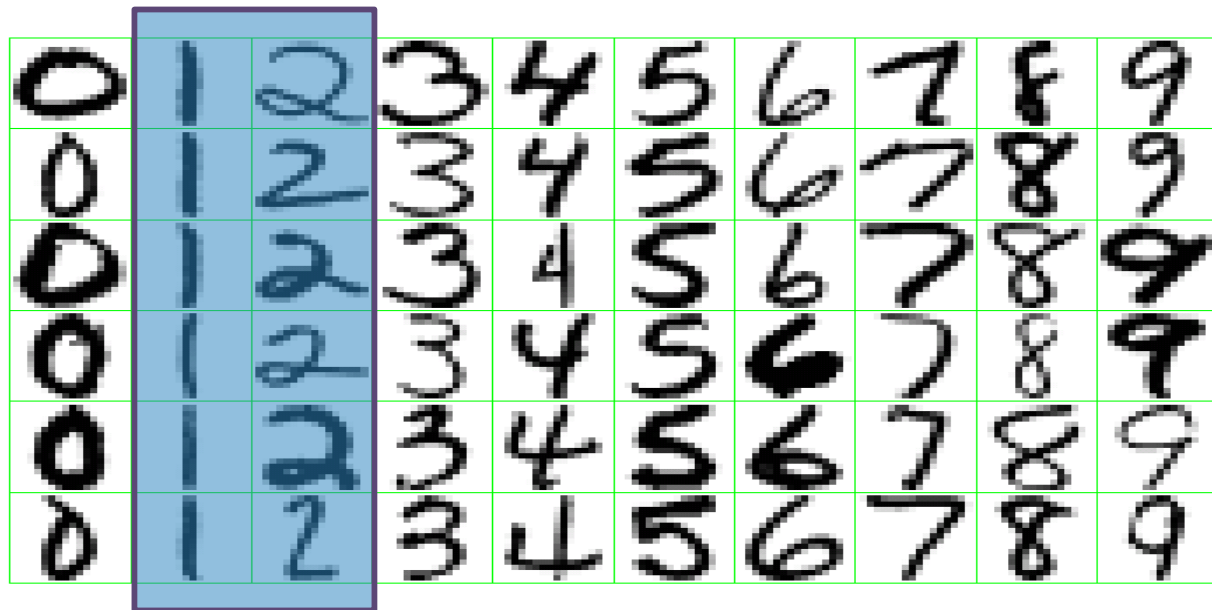
- Logistic regression is a "linear classifier"
- Example in 2-D
 - $D(\theta, x) = (\theta_0 + \theta_1 x_1 + \theta_2 x_2)$
 - $\theta^T = [-3, 1, 1]$
 - $\theta^T x = -3 + x_1 + x_2 \geq 0$
 - $\rightarrow x_1 + x_2 \geq 3$

Decision Boundary: Non-Linear

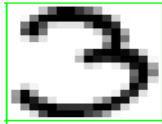


- “Non-linear” algorithms have more complicated decision boundaries
- $D(\theta, x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$
- $\theta^T = [-2, 0, 0, 1, 1]$
- $\rightarrow x_1^2 + x_2^2 \geq 2$

Example: Handwritten Digit Recognition



Extracting Features For Learning



$\{x_1, x_2, x_3, \dots, x_{256}, y = \text{'three'}\}$

- Each x_i corresponds to a feature value in the image
- y is a label of the training data; can be numeric or categorical, '3' or 'three'
- Each image is converted to row vectors and the appropriate learning algorithm is used
- Convention
 - x_i represents the i th feature in a training sample
 - y represents the label for the training sample

QUESTIONS