Logistic Regression

Classification

WEEK 4, COHORT 4

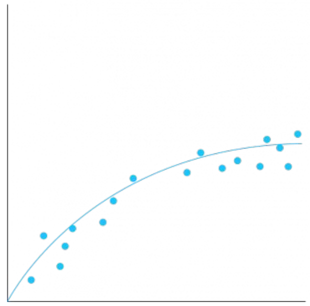
Supervised Learning:

Predicting values. **Known** targets.
User inputs correct answers to learn from. Machine uses the information to guess new answers.

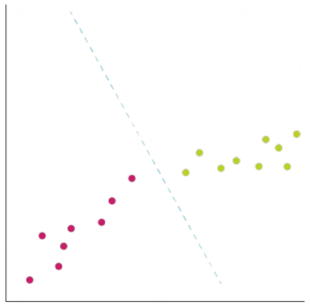
REGRESSION:
Estimate continuous values
(Real-valued output)

CLASSIFICATION:
Identify a unique class
(Discrete values, Boolean, Categories)

Regression



Classification



Unsupervised Learning:

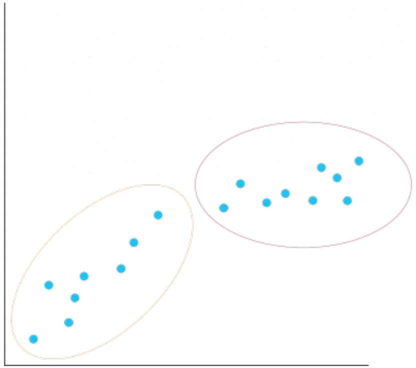
Search for structure in data. **Unknown** targets.
User inputs data with undefined answers. Machine finds useful information hidden in data.

Cluster Analysis
Group into sets

Density Estimation
Approximate distributions

Dimension Reduction
Select relevant variables

Clustering



Classification

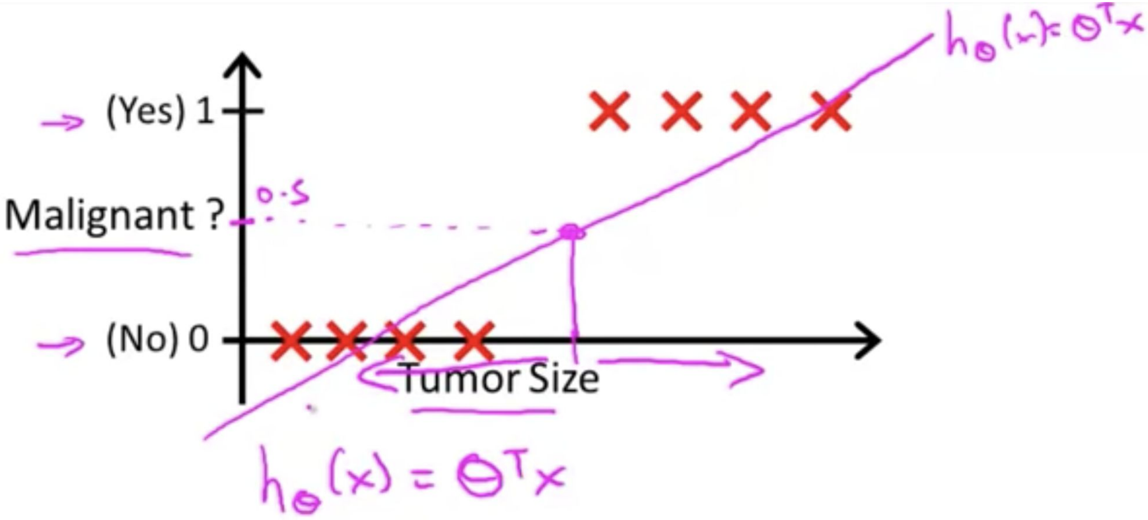
- Email: Spam / Not Spam?
- Online Transactions: Fraudulent (Yes / No)?
- Tumor: Malignant / Benign ?

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

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

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

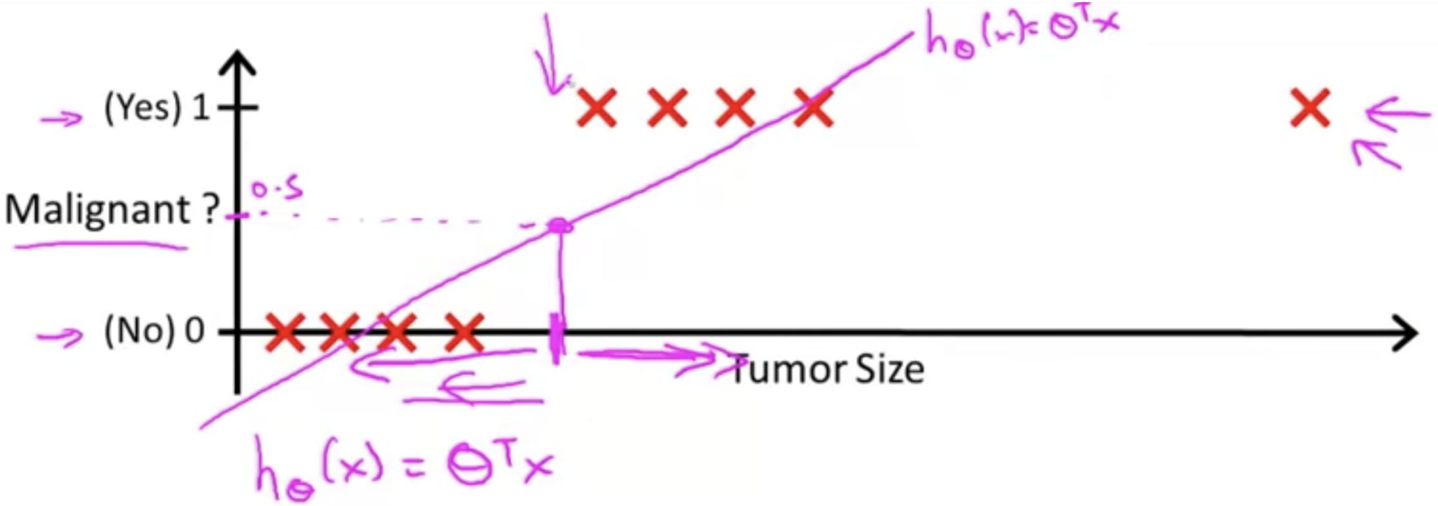
$y \in \{0, 1, 2, 3\}$



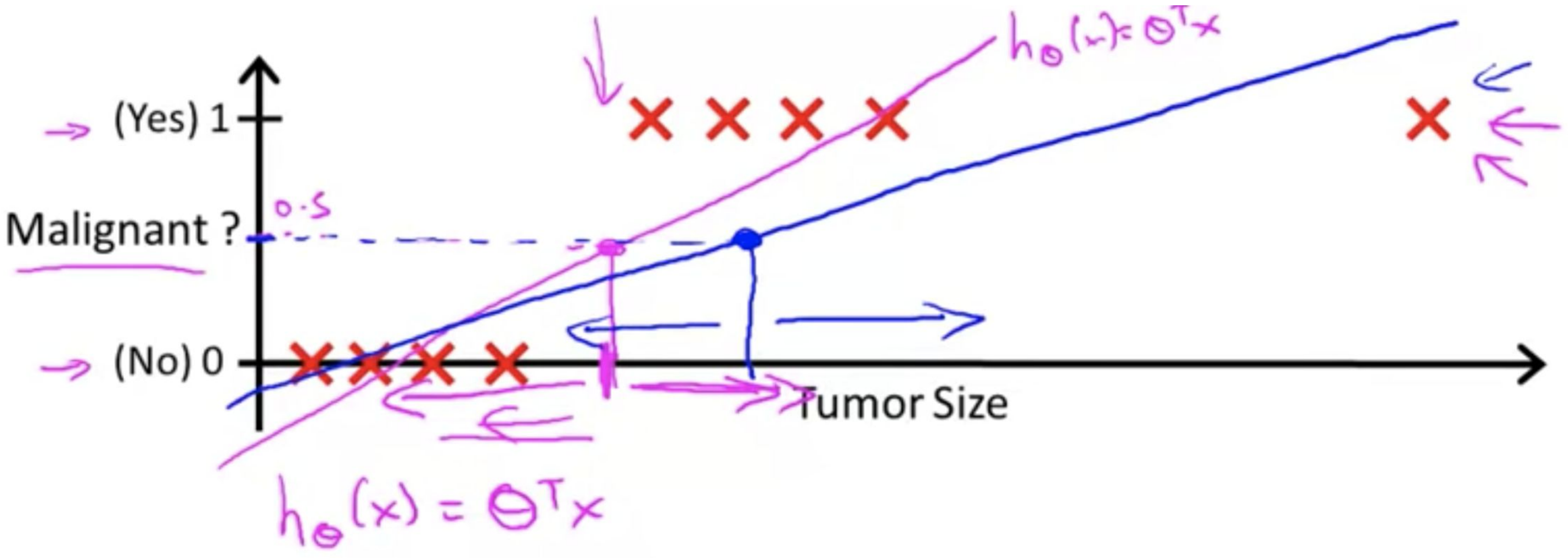
Threshold classifier output $h_\theta(x)$ at 0.5:

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"



Applying linear regression to classification problem is a bad idea!

Classification: $\boxed{y = 0 \text{ or } 1}$

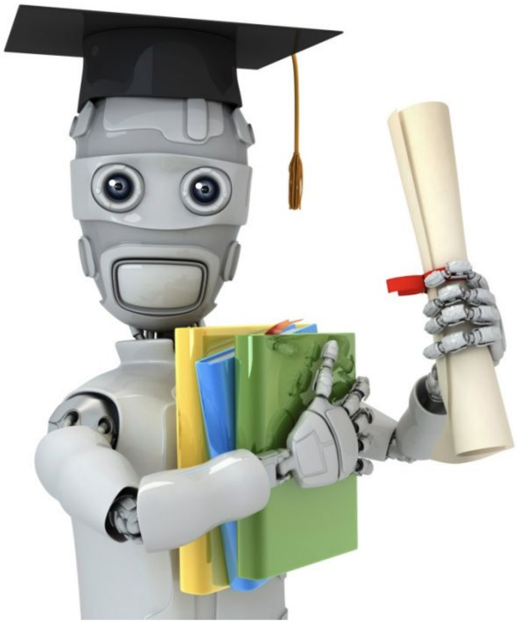
$h_\theta(x)$ can be > 1 or < 0

Why linear regression is bad for classification.

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

Classification

What we want



Machine Learning

Logistic Regression

Hypothesis Representation

Logistic Regression Model

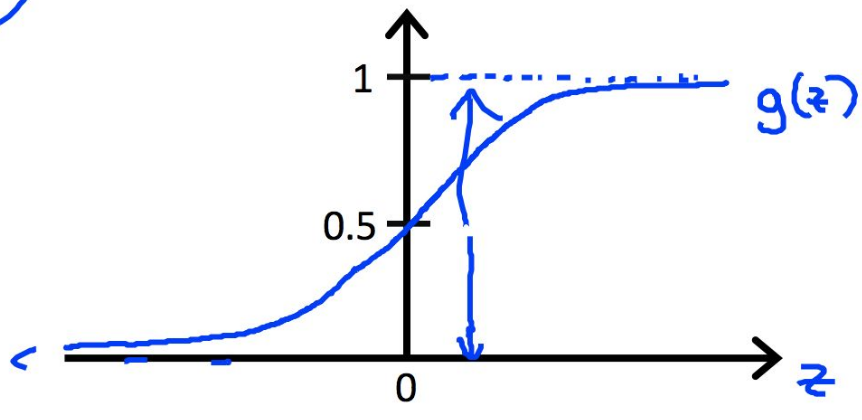
Want $\boxed{0 \le h_\theta(x) \le 1}$

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

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

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

$\theta^T x$



Sigmoid function

Logistic function

Parameters θ.

Interpretation of Hypothesis Output

$h_\theta(x)$

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

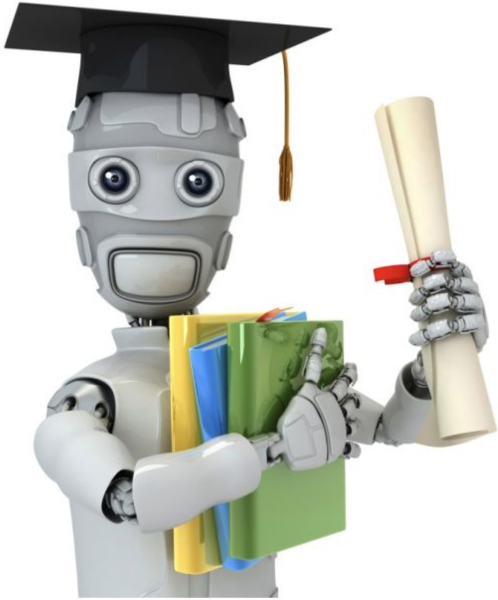
$h_\theta(x) = P(y=1|x;\theta)$

$y = 0$ or 1

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

$$P(y=0|\cancel{x;\theta}) + \boxed{P(y=1|\cancel{x;\theta})} = 1$$

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



Machine Learning

Logistic Regression

Decision boundary

DECISION BOUNDARY

The hypothesis gives values ranging from 0 to 1. In order to obtain the class of the prediction, we can establish a **threshold** in which values greater than the threshold are rounded up as 1 and lesser values are 0. We can choose 0.5

$$h_\theta(x) \geq 0.5 \rightarrow y = 1$$
$$h_\theta(x) < 0.5 \rightarrow y = 0$$

The logistic/sigmoid function has a very interesting behaviour. When its input is greater than or equal to 0, its output is greater than or equal to 0.5, and when its input is less than 0, its output is less than 0.5.

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

$$\theta^T x \geq 0 \Rightarrow y = 1$$
$$\theta^T x < 0 \Rightarrow y = 0$$

The **decision boundary** is a line that separates the areas y=1 and y=0

$$\theta = \begin{bmatrix} 5 \\ -1 \\ 0 \end{bmatrix}$$

$$y = 1 \; if \; 5 + (-1)x_1 + 0x_2 \geq 0$$

$$5 - x_1 \geq 0$$

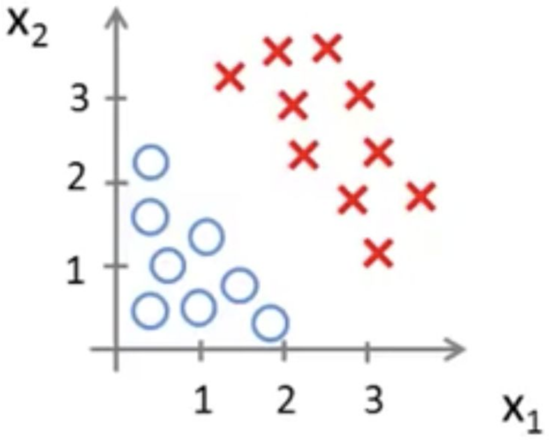
$$-x_1 \geq -5$$

$$x_1 \leq 5$$

Decision boundary

Find the decision boundary?

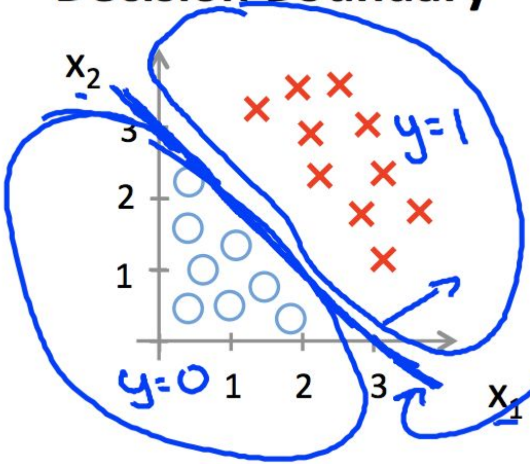
Decision Boundary



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

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

Decision Boundary



Decision boundary

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

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

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

$\rightarrow x_1 + x_2 \geq 3$

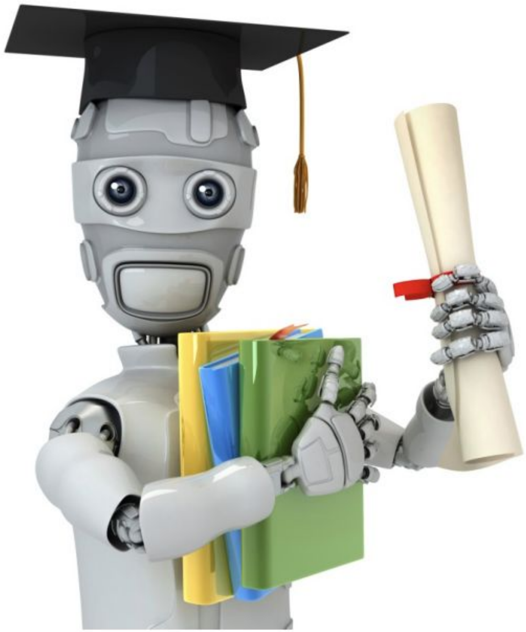
x_1, x_2

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

$\boxed{x_1 + x_2 = 3}$

$x_1 + x_2 < 3$

$\rightarrow y = 0$



Machine Learning

Logistic Regression

Cost function

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

m examples

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

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

How to choose parameters θ ?

A **cost function** tells us **“how good”** our model is at making predictions for a given set of parameters. The cost function has its own curve and its own gradients. The slope of this curve tells us how to update our parameters to make the model more accurate.

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Questions:

1 - What cost function should we use ?

2 - Should we use MSE as in Linear regression ?

COST FUNCTION FOR LOGISTIC REGRESSION

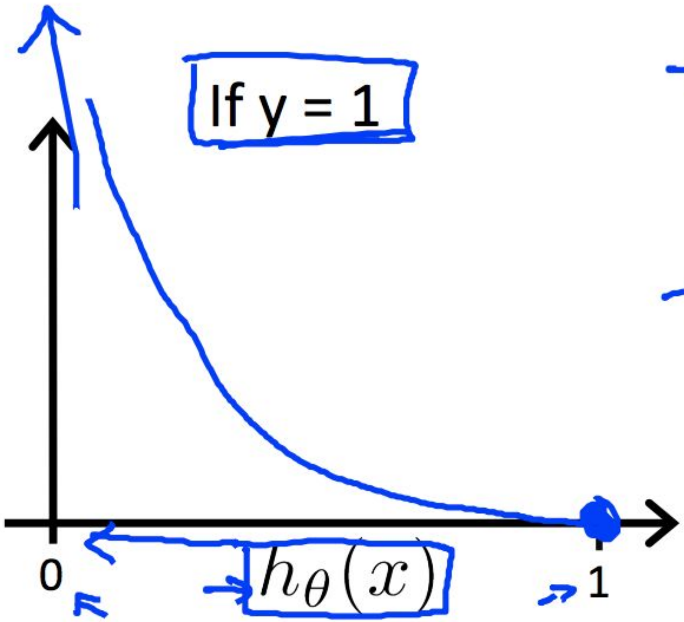
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

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

LOGISTIC REGRESSION: COST FUNCTION

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



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

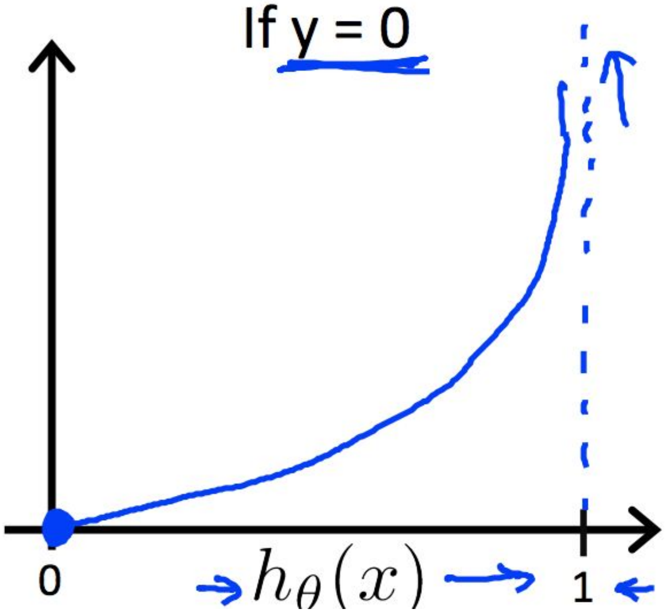
But as $h_\theta(x) \to 0$

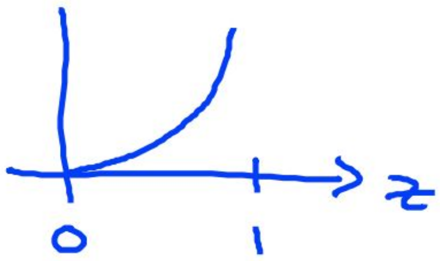
$Cost \to \infty$

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

LOGISTIC REGRESSION: COST FUNCTION

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

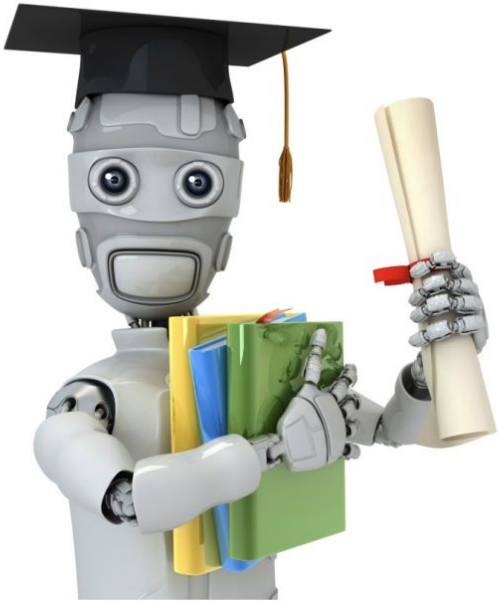




SUMMARY OF COST FUNCTION

If our hypothesis is far from y, the cost tends to infinity.
If our hypothesis is exactly the same as y, the cost is zero.

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



Logistic Regression

Simplified cost function and gradient descent

Machine Learning

Simplified Cost Function and Gradient Descent

We can write the cost function's conditional cases in a more compact form

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

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

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

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

for $i = 0$ to n

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

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

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

Algorithm looks identical to linear regression!

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

Hinton's Closing Prayer

Our father who art in n-dimensions

hallowed by the backprop,

thy loss be minimized,

thy gradients unvarnished,

on earth as it is in Euclidean space.

Give us this day our daily hyperparameters,

and forgive us our large learning rates,

as we forgive those whose parameters diverge,

and lead us not into discrete optimization,

but deliver us from local optima.

For thine are dimensions,

and the GPUs, and the glory,

forever and ever. Dropout.

From buZZrobot