

Logistic Regression

Prof. Dr. Christina Bauer

christina.bauer@th-deg.de

Faculty of Computer Science

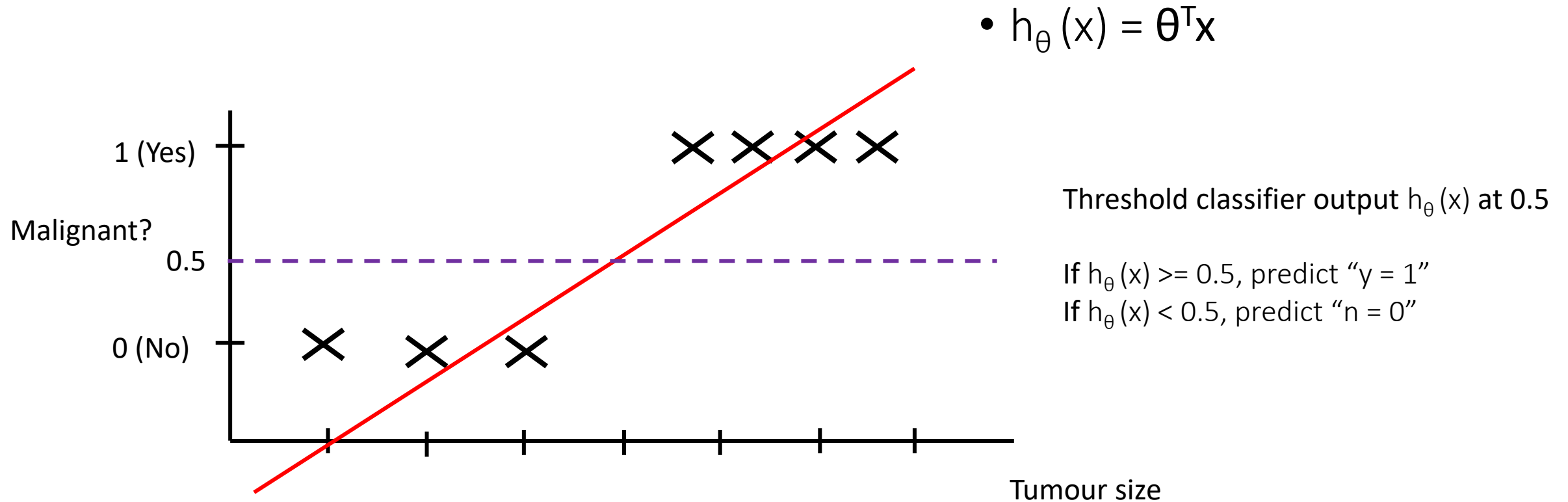
CLASSIFICATION

- Email: Spam/not Spam
- Online transactions: Fraudulent (yes/no)
- Tumour: malignant/benign

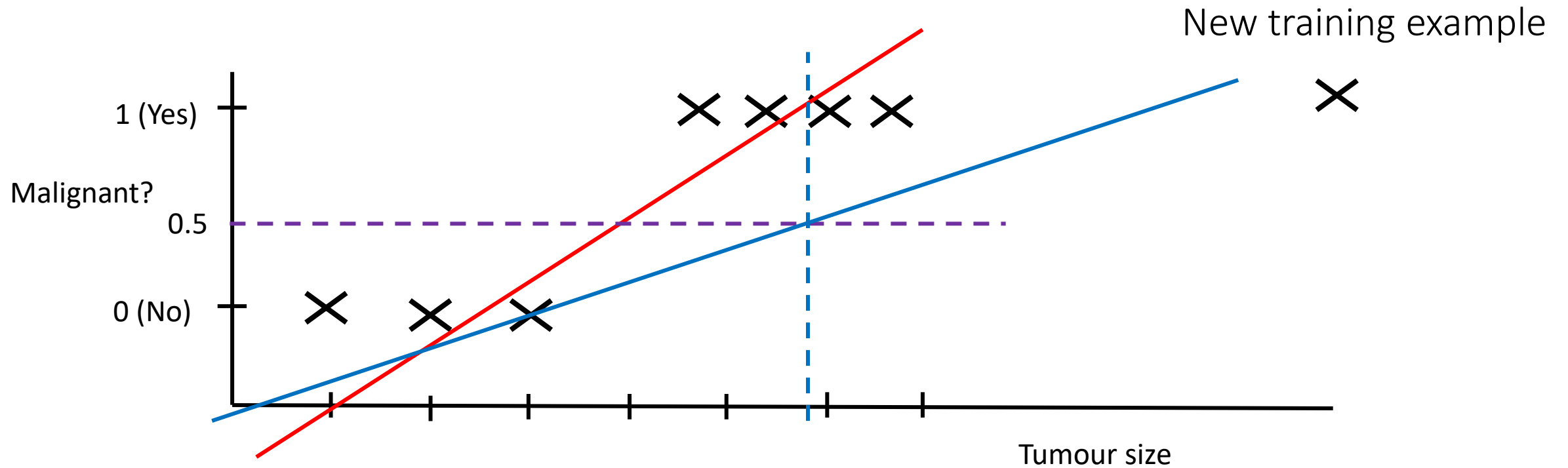
$y \in \{0,1\}$ 0: “negative class” (e. g. benign tumour)
 1: “positive class” (e. g. malignant tumour)

- Multiclass classification problem $\rightarrow y \in \{0,1,2,3\}$

CLASSIFICATION WITH LINEAR REGRESSION



CLASSIFICATION WITH LINEAR REGRESSION

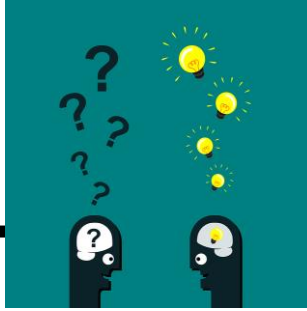


CLASSIFICATION WITH LINEAR REGRESSION

- Classification $y = 0$ or 1
- $h_{\theta}(x)$ can be >1 or <0

→ Logistic Regression $0 \leq h_{\theta}(x) \leq 1$

QUESTION



Which of the following statements is true?

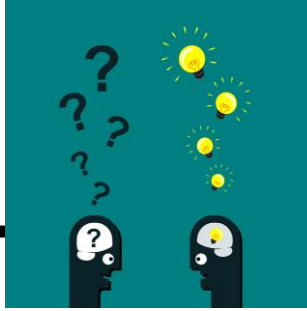
A: If linear regression doesn't work on a classification task as in the previous example applying feature scaling may help.

B: If the training set satisfies $0 \leq y^{(i)} \leq 1$ for every training example $(x^{(i)}, y^{(i)})$ then linear regression's prediction will also satisfy $0 \leq h_{\theta}(x) \leq 1$ for all values of x .

C: If there is a feature x that perfectly predicts y , i. e. if $y = 1$ when $x \geq c$ and $y = 0$ whenever $x < c$ (for some constant c), then linear regression will obtain zero classification error.

D: None of the above statements are true.

QUESTION



Which of the following statements is true?

A: If linear regression doesn't work on a classification task as in the previous example applying feature scaling may help.

B: If the training set satisfies $0 \leq y^{(i)} \leq 1$ for every training example $(x^{(i)}, y^{(i)})$ then linear regression's prediction will also satisfy $0 \leq h_{\theta}(x) \leq 1$ for all values of x .

C: If there is a feature x that perfectly predicts y , i. e. if $y = 1$ when $x \geq c$ and $y = 0$ whenever $x < c$ (for some constant c), then linear regression will obtain zero classification error.

~~D~~: None of the above statements are true.

HYPOTHESIS REPRESENTATION

Logistic Regression Model

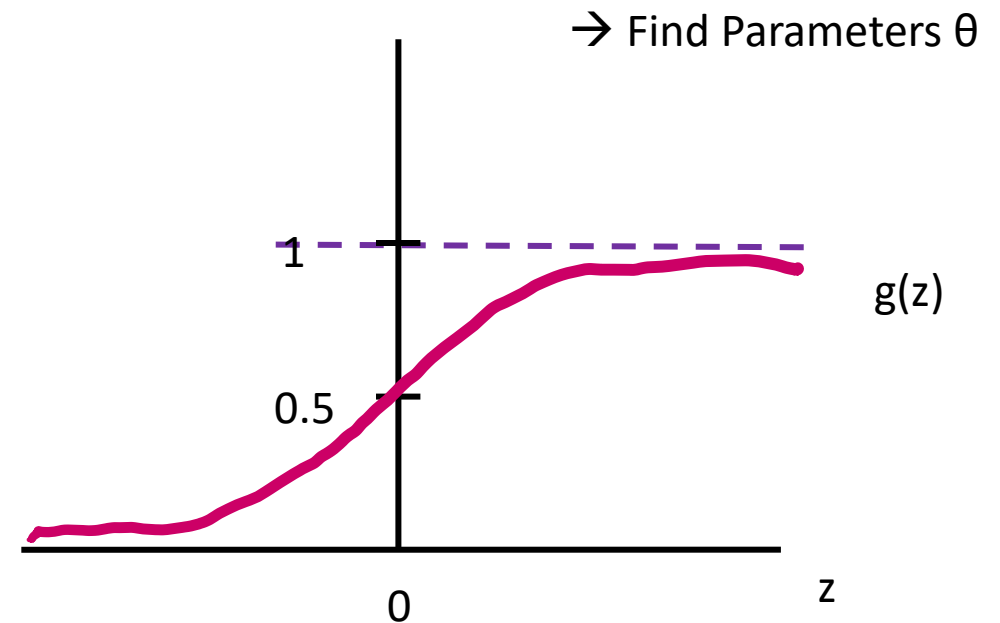
$0 \leq h_{\theta}(x) \leq 1 \rightarrow$ classifier should output values between 0 and 1

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

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

Sigmoid function
Logistic function

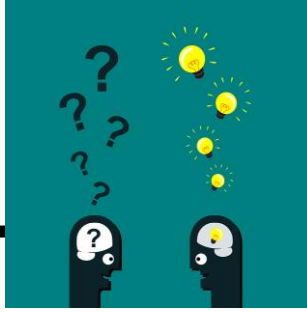
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$



INTERPRETATION OF HYPOTHESIS OUTPUT

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on output x
- Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{Tumour size} \end{bmatrix}$
- $h_{\theta}(x) = 0.7$
- Tell patient that 70 % chance of tumour being malignant
- $h_{\theta}(x) = p(y=1 | x; \theta) \rightarrow$ Probability that $y = 1$, given x , parametrized by θ

QUESTION



Suppose we want to predict, from data x about a tumour, whether it is malignant ($y = 1$) or benign ($y = 0$). Our logistic regression classifier outputs, for a specific tumour, $h_{\theta}(x) = p(y=1|x;\theta) = 0.7$, so we estimate that there is a 70 % chance of this tumour being malignant. What should be our estimate for $P(y=0|x;\theta)$, the probability the tumour is benign?

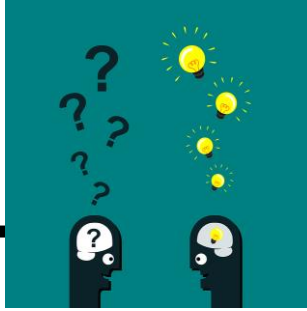
A: $P(y=0|x;\theta) = 0.3$

B: $P(y=0|x;\theta) = 0.7$

C: $P(y=0|x;\theta) = 0.7^2$

D: $P(y=0|x;\theta) = 0.3 \times 0.7$

QUESTION



Suppose we want to predict, from data x about a tumour, whether it is malignant ($y = 1$) or benign ($y = 0$). Our logistic regression classifier outputs, for a specific tumour, $h_{\theta}(x) = p(y=1|x;\theta) = 0.7$, so we estimate that there is a 70 % chance of this tumour being malignant. What should be our estimate for $P(y=0|x;\theta)$, the probability the tumour is benign?

~~A~~: $P(y=0|x;\theta) = 0.3$

B: $P(y=0|x;\theta) = 0.7$

C: $P(y=0|x;\theta) = 0.7^2$

D: $P(y=0|x;\theta) = 0.3 \times 0.7$

INTERPRETATION OF HYPOTHESIS OUTPUT

- $h_{\theta}(x)$ = estimated probability that $y = 1$ on output x
 - Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{Tumour size} \end{bmatrix}$
 - $h_{\theta}(x) = 0.7$
 - Tell patient that 70 % chance of tumour being malignant
 - $h_{\theta}(x) = p(y=1 | x; \theta) \rightarrow$ Probability that $y = 1$, given x , parametrized by θ
- $\rightarrow p(y=1 | x; \theta) + p(y=0 | x; \theta) = 1$
- $\rightarrow p(y=0 | x; \theta) = 1 - p(y=1 | x; \theta)$

DECISION BOUNDARY

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

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

Suppose predict “y = 1” if $h_{\theta}(x) \geq 0.5$

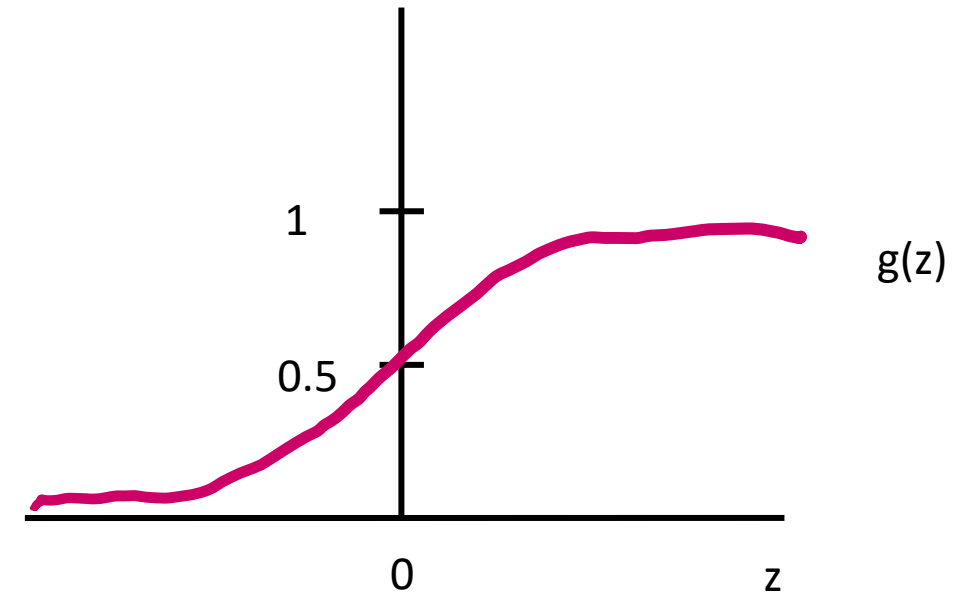
$$g(z) \geq 0.5 \text{ when } z \geq 0$$

$$h_{\theta}(x) = g(\theta^T x) \geq 0.5 \text{ when } \theta^T x \geq 0$$

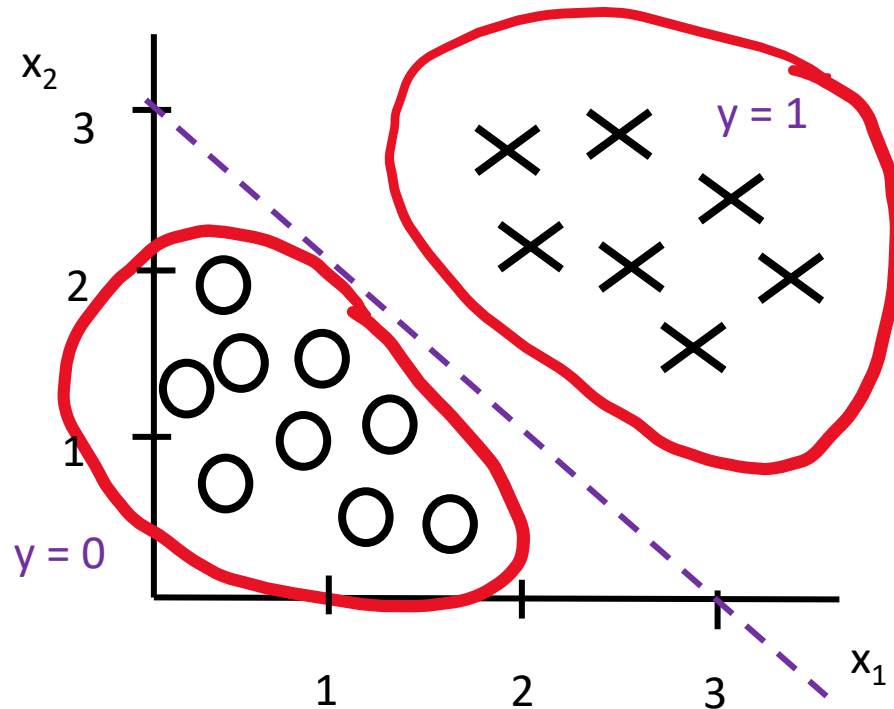
Predict “y = 0” if $h_{\theta}(x) < 0.5$

$$g(z) < 0.5 \text{ when } z < 0$$

$$h_{\theta}(x) = g(\theta^T x) < 0.5 \text{ when } \theta^T x < 0$$



DECISION BOUNDARY



Decision Boundary

$$x_1 + x_2 = 3$$

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

\rightarrow Is a property of the hypothesis not of the data set

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

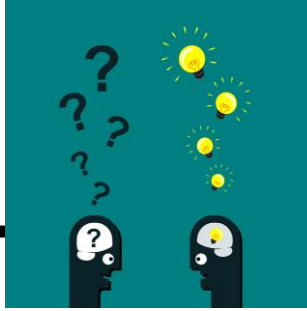
- $\Theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$

- Predict “y=1” $\theta^T x \geq 0$

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

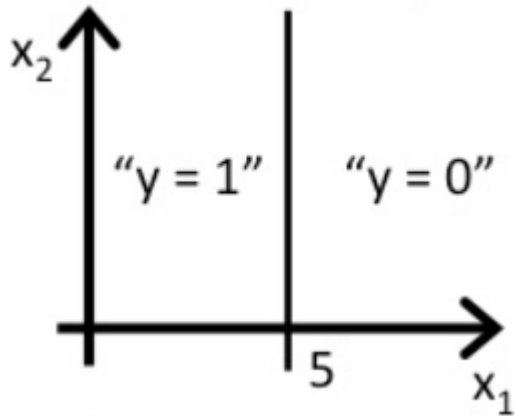
- Predict “y=1” if $x_1 + x_2 \geq 3$

QUESTION

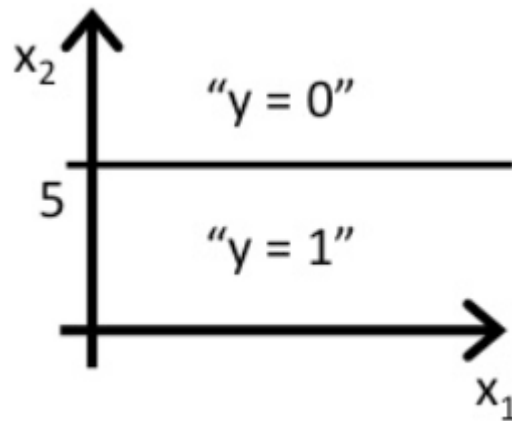


Consider logistic regression with two features x_1 and x_2 . Suppose $\theta_0 = 5$, $\theta_1 = -1$, $\theta_2 = 0$, so that $h_{\theta}(x) = g(5 - x_1)$. Which of these shows the decision boundary of $h_{\theta}(x)$?

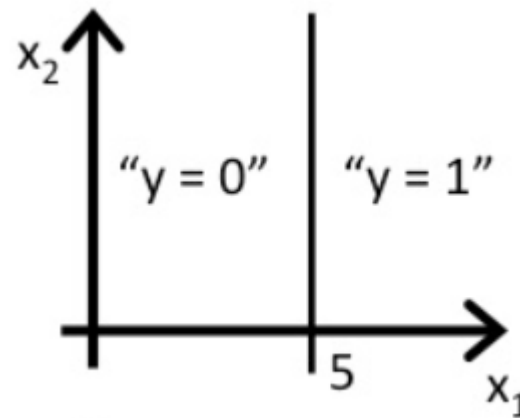
A



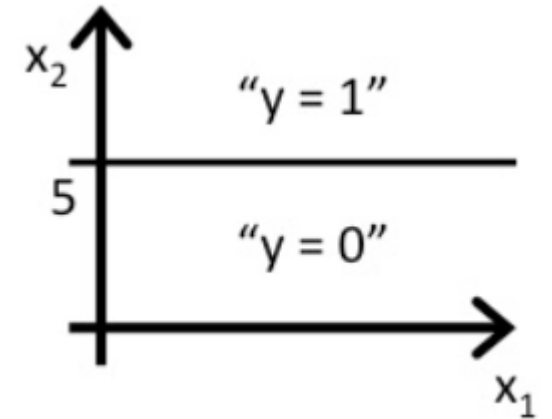
B



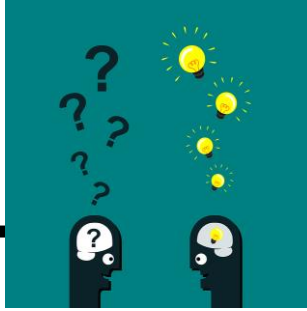
C



D

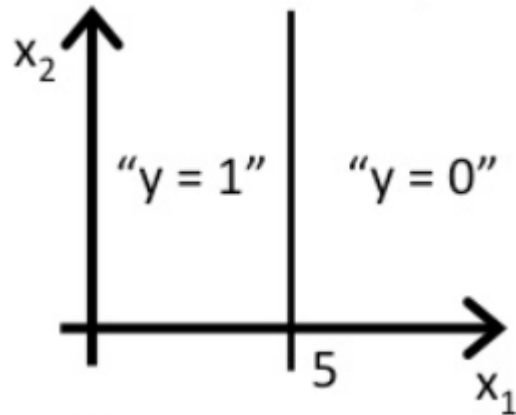


QUESTION

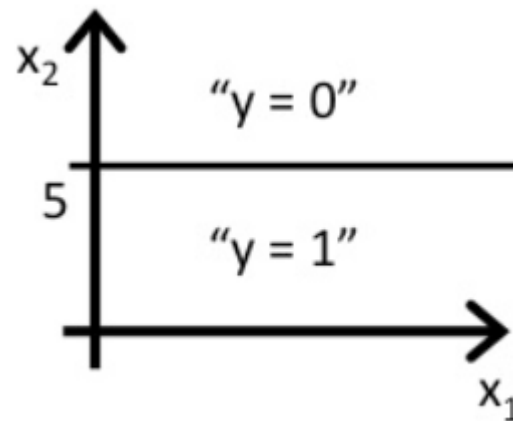


Consider logistic regression with two features x_1 and x_2 . Suppose $\theta_0 = 5$, $\theta_1 = -1$, $\theta_2 = 0$, so that $h_\theta(x) = g(5 - x_1)$. Which of these shows the decision boundary of $h_\theta(x)$?

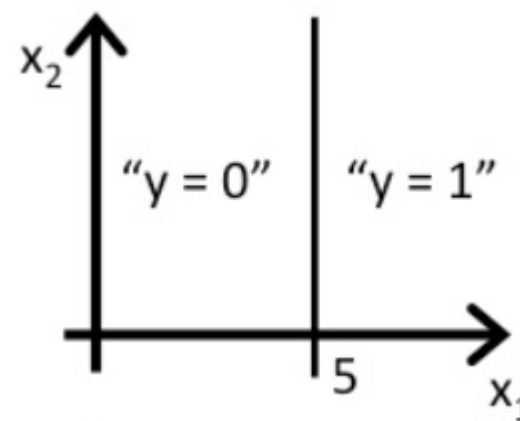
~~A~~



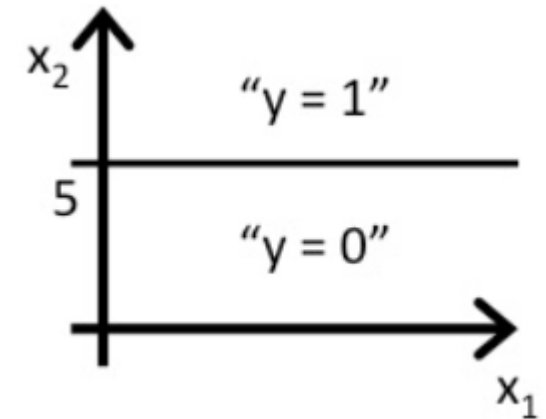
B



C

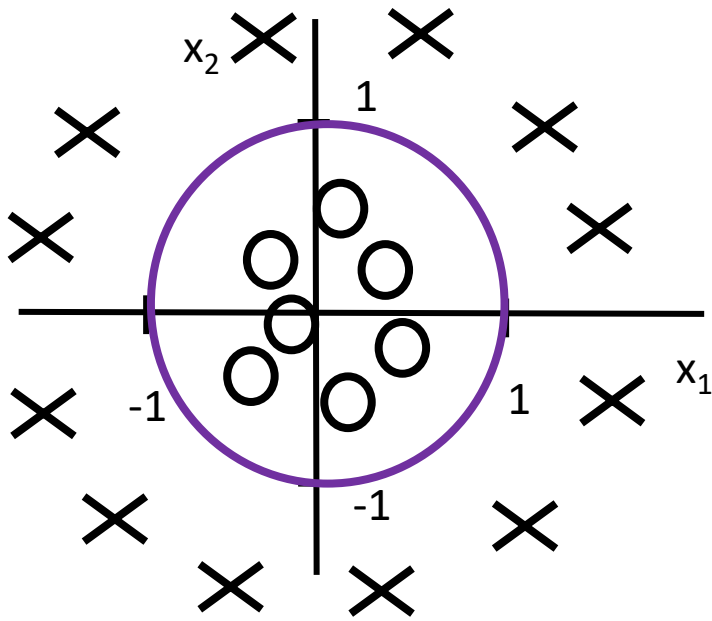


D



\rightarrow "y=1" if $5 - x_1 \geq 0 \rightarrow$ "y=1" if $x_1 \leq 5$

NON-LINEAR DECISION BOUNDARIES



Decision Boundary
 $(x_1)^2 + (x_2)^2 = 1$

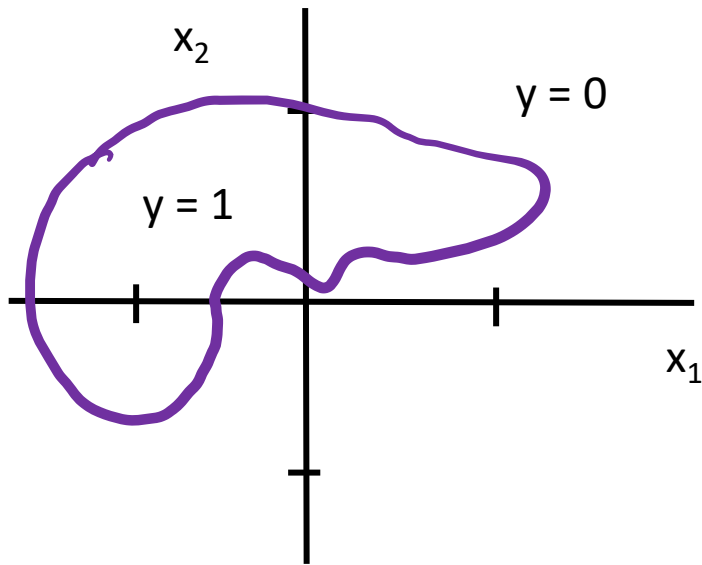
- $h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 (x_1)^2 + \theta_4 (x_2)^2)$

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

- Predict “y=1” if $-1 + (x_1)^2 + (x_2)^2 \geq 0$

- Predict “y=1” if $(x_1)^2 + (x_2)^2 \geq 1$

NON-LINEAR DECISION BOUNDARIES



Decision Boundary

- complex decision boundaries are possible
- $$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 (x_1)^2 + \theta_4 (x_2)^2 x_2 + \theta_5 (x_1)^2 (x_2)^2 + \theta_6 (x_1)^3 x_2 + \dots)$$

SUPERVISED LEARNING PROBLEM – LOGISTIC REGRESSION MODEL

Training Set: $((x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)}))$

m examples $x \in \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}$ $x_0 = 1; y \in \{0, 1\}$

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

How to choose parameters θ ?

COST FUNCTION

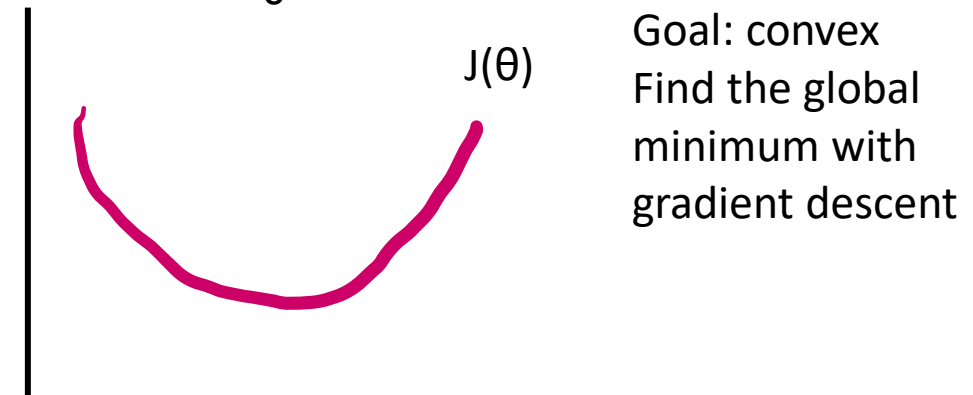
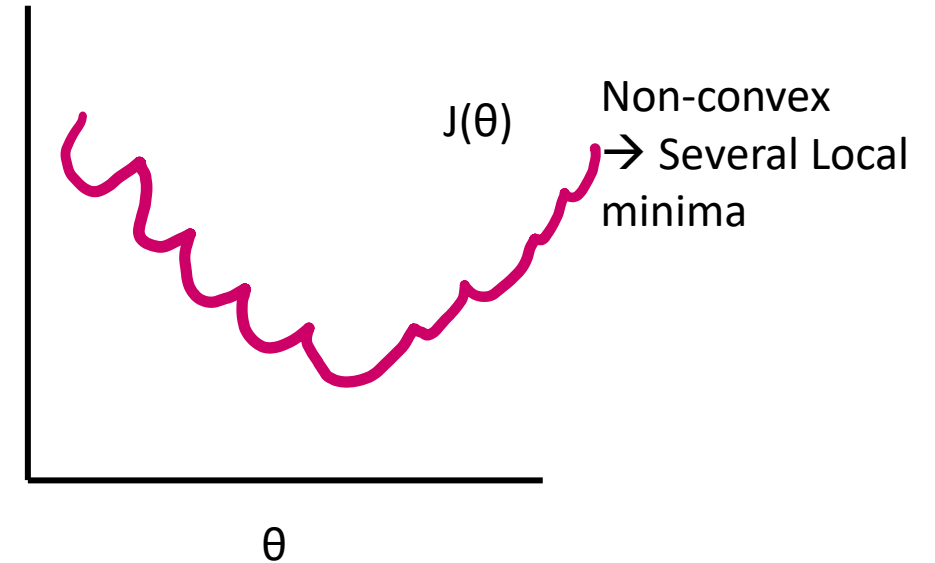
- Linear Regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2$
- Alternative: $J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{cost}(h_{\theta}(x^{(i)}), y^{(i)})$
- Cost $(h_{\theta}(x^{(i)}), y^{(i)}) = \frac{1}{2} (h_{\theta}(x^{(i)}) - y^{(i)})^2$
- Simplify: $\text{Cost}(h_{\theta}(x), y) = \frac{1}{2} (h_{\theta}(x) - y)^2$
- Suitable for Logistic Regression?

COST FUNCTION

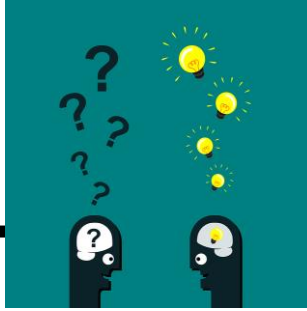
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{cost}(h_{\theta}(x^{(i)}), y^{(i)})$$

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

$$\bullet h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}} \quad \text{Non-linear function}$$

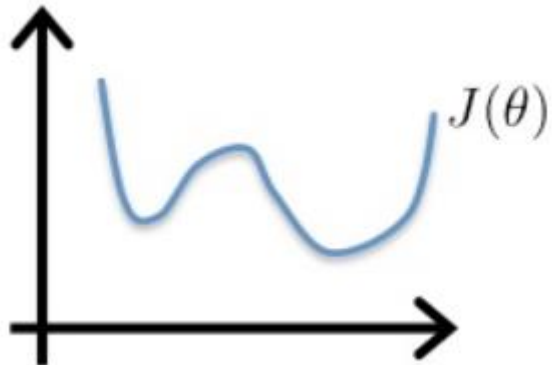


QUESTION

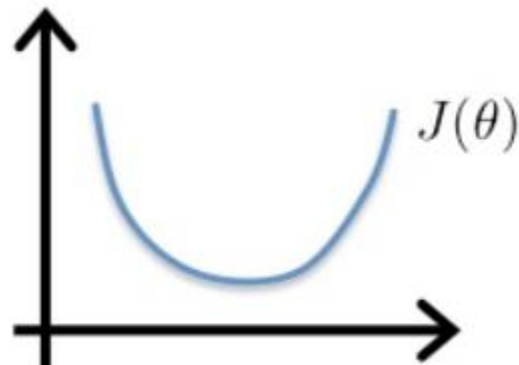


Consider minimizing a cost function $J(\theta)$. Which one of these functions is convex?

A



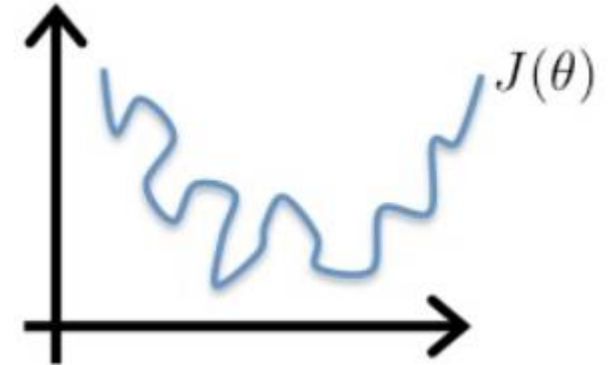
B



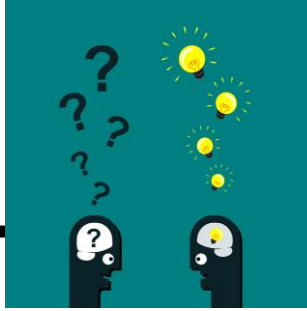
C



D

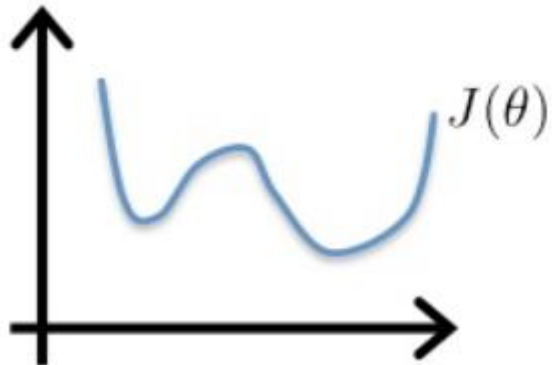


QUESTION

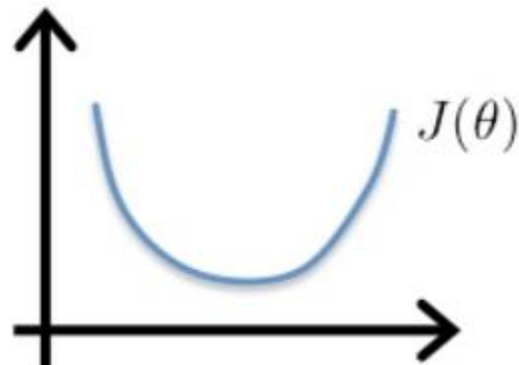


Consider minimizing a cost function $J(\theta)$. Which one of these functions is convex?

A



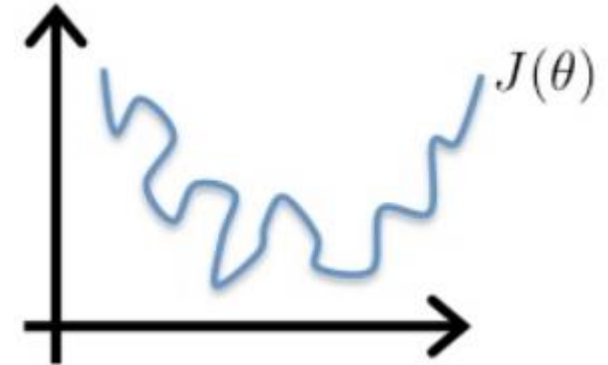
~~B~~



C

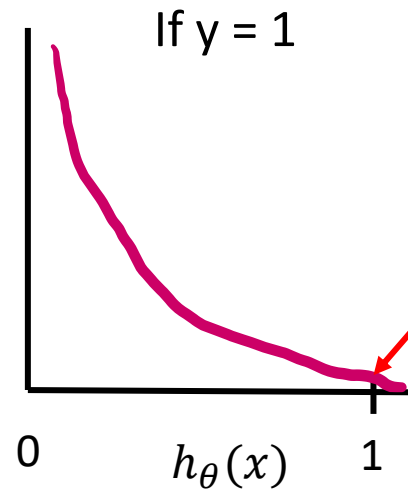


D



LOGISTIC REGRESSION COST FUNCTION

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



Cost = 0 if $y = 1$, $h_{\theta}(x) = 1$ (i. e. “best match”)

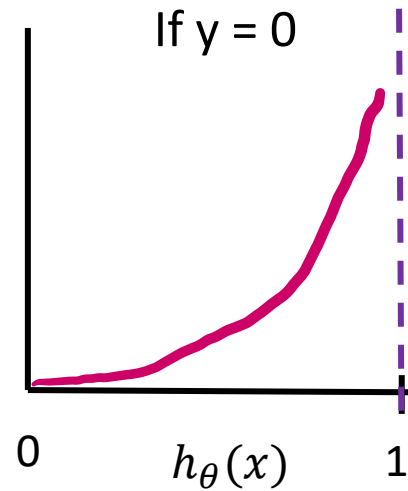
But as $h_{\theta}(x) \rightarrow 0$

Cost $\rightarrow \infty$

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

LOGISTIC REGRESSION COST FUNCTION

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

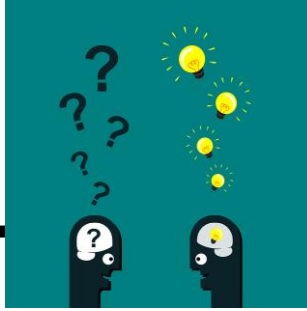


Cost = 0 if $y = 0$, $h_{\theta}(x) = 0$ (i. e. “best match”)

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

Cost $\rightarrow \infty$

QUESTION



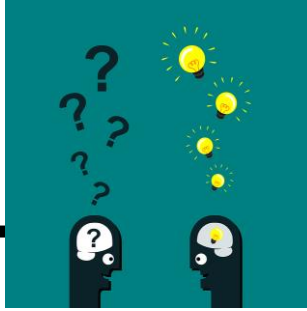
In logistic regression, the cost function for our hypothesis outputting (predicting) $h_{\theta}(x)$ on a training example that has label $y \in \{0,1\}$ is:

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

Which of the following are true? Check all that apply.

- A: If $h_{\theta}(x) = y$, then $\text{cost}(h_{\theta}(x), y) = 0$ (for $y = 0$ and $y = 1$)
- B: If $y = 0$, then $\text{cost}(h_{\theta}(x), y) \rightarrow \infty$ as $h_{\theta}(x) \rightarrow 1$.
- C: If $y = 0$, then $\text{cost}(h_{\theta}(x), y) \rightarrow \infty$ as $h_{\theta}(x) \rightarrow 0$.
- D: Regardless of whether $y = 0$ or $y = 1$, if $h_{\theta}(x) = 0.5$ then $\text{cost}(h_{\theta}(x), y) > 0$.

QUESTION



In logistic regression, the cost function for our hypothesis outputting (predicting) $h_\theta(x)$ on a training example that has label $y \in \{0,1\}$ is:

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

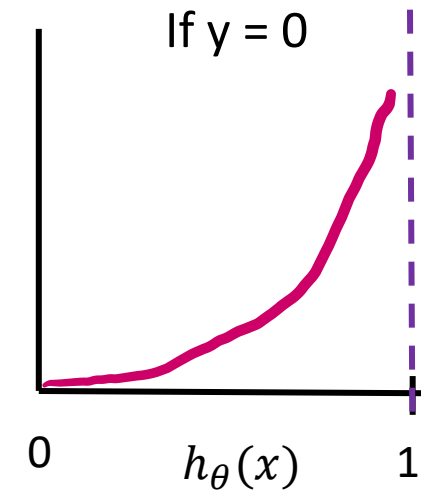
Which of the following are true? Check all that apply.

☒ A: If $h_\theta(x) = y$, then $\text{cost}(h_\theta(x), y) = 0$ (for $y = 0$ and $y = 1$)

☒ B: If $y = 0$, then $\text{cost}(h_\theta(x), y) \rightarrow \infty$ as $h_\theta(x) \rightarrow 1$.

☐ C: If $y = 0$, then $\text{cost}(h_\theta(x), y) \rightarrow \infty$ as $h_\theta(x) \rightarrow 0$.

☒ D: Regardless of whether $y = 0$ or $y = 1$, if $h_\theta(x) = 0.5$ then $\text{cost}(h_\theta(x), y) > 0$.



LOGISTIC REGRESSION COST FUNCTION

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

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

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

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

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

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

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

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 :

$\min_{\theta} J(\theta) \rightarrow$ gives us θ

To make a prediction given new x :

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

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)


}

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

GRADIENT DESCENT

$$J(\theta) = y \log(h_{\theta}(x)) + (1-y) \log(1-h_{\theta}(x))$$

$$\frac{\partial}{\partial \theta} J(\theta) = y \frac{1}{h_{\theta}(x)} * \frac{\partial h_{\theta}(x)}{\partial \theta} + (1-y) * \frac{1}{1-h_{\theta}(x)} * \frac{\partial(1-h_{\theta}(x))}{\partial \theta}$$

$$\frac{\partial h_{\theta}(x)}{\partial \theta} = h_{\theta}(x) * x * (1-h_{\theta}(x))$$


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

GRADIENT DESCENT

$$\frac{\partial}{\partial \theta} J(\theta) = y \frac{1}{h_{\theta}(x)} * \frac{\partial h_{\theta}(x)}{\partial \theta} + (1 - y) * \frac{1}{1 - h_{\theta}(x)} * \frac{\partial (1 - h_{\theta}(x))}{\partial \theta}$$

$$\frac{\partial}{\partial \theta} J(\theta) = y \frac{1}{h_{\theta}(x)} * [h_{\theta}(x) * x * (1 - h_{\theta}(x))] + (1 - y) * \frac{1}{1 - h_{\theta}(x)} * \{0 - [h_{\theta}(x) * x * (1 - h_{\theta}(x))]\}$$

$$\frac{\partial}{\partial \theta} J(\theta) = y * x * (1 - h_{\theta}(x)) + (1 - y) * \frac{1}{1 - h_{\theta}(x)} * \{-[h_{\theta}(x) * x * (1 - h_{\theta}(x))]\}$$

$$\frac{\partial}{\partial \theta} J(\theta) = y * x * (1 - h_{\theta}(x)) + (1 - y) * -h_{\theta}(x) * x$$

$$\frac{\partial}{\partial \theta} J(\theta) = y * x - h_{\theta}(x) * y * x - h_{\theta}(x) * x + y * x * h_{\theta}(x) = y * x - h_{\theta}(x) * x = (y - h_{\theta}(x)) * x$$

GRADIENT DESCENT

$$\frac{\partial}{\partial \theta} J(\theta) = (y - h_{\theta}(x)) * x$$

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

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

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 {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j$$

(simultaneously update all θ_j)

}

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

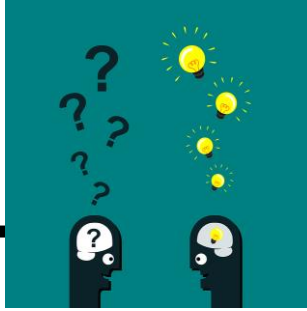
Linear Regression: $h_{\theta}(x^{(i)}) = \theta^T x$

Logistic Regression:

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

→ Algorithm looks identical to linear regression

QUESTION



Suppose you are running gradient descent to fit a logistic regression model with parameter $\theta \in \mathbb{R}^{n+1}$. Which of the following is a reasonable way to make sure the learning rate α is set properly and that gradient descent is running correctly?

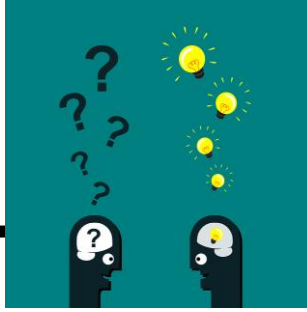
A: Plot $J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$ as a function of the number of iterations (i.e. the horizontal axis is the iteration number) and make sure $J(\theta)$ is decreasing on every iteration.

B: Plot $J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$ as a function of the number of iterations and make sure $J(\theta)$ is decreasing on every iteration.

C: Plot $J(\theta)$ as a function of θ and make sure it is decreasing on every iteration.

D: Plot $J(\theta)$ as a function of θ make sure it is convex.

QUESTION



Suppose you are running gradient descent to fit a logistic regression model with parameter $\theta \in \mathbb{R}^{n+1}$. Which of the following is a reasonable way to make sure the learning rate α is set properly and that gradient descent is running correctly?

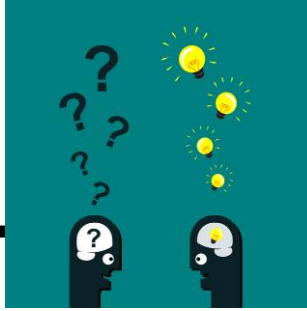
A: Plot $J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$ as a function of the number of iterations (i.e. the horizontal axis is the iteration number) and make sure $J(\theta)$ is decreasing on every iteration.

~~B~~: Plot $J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$ as a function of the number of iterations and make sure $J(\theta)$ is decreasing on every iteration.

C: Plot $J(\theta)$ as a function of θ and make sure it is decreasing on every iteration.

D: Plot $J(\theta)$ as a function of θ make sure it is convex.

QUESTION



One iteration of gradient descent simultaneously performs these updates:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_1$$

...

$$\theta_n := \theta_n - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_n$$

We would like a vectorized implementation of the form $\theta := \theta - \alpha \delta$ (for some vector $\delta \in \mathbb{R}^{n+1}$).

What should the vectorized implementation be?

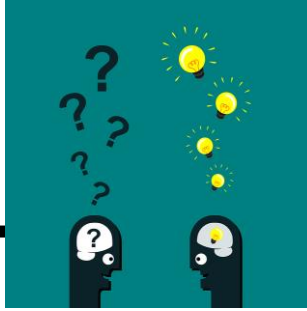
A: $\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}]$

B: $\theta := \theta - \alpha \frac{1}{m} [\sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)})]] * x^{(i)}$

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

D: All of the above are correct implementations.

QUESTION



One iteration of gradient descent simultaneously performs these updates:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_1$$

...

$$\theta_n := \theta_n - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_n$$

We would like a vectorized implementation of the form $\theta := \theta - \alpha \delta$ (for some vector $\delta \in \mathbb{R}^{n+1}$).

What should the vectorized implementation be?

~~A: $\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}]$~~

B: $\theta := \theta - \alpha \frac{1}{m} [\sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)})]] * x^{(i)}$

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

D: All of the above are correct implementations.

ADVANCED OPTIMIZATION

- Cost function $J(\theta) \rightarrow \min_{\theta} J(\theta)$
- Given θ , we need code that can compute

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

Repeat {

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

}

ADVANCED OPTIMIZATION

Given θ , we need code that can compute

$$J(\theta)$$
$$\frac{\partial}{\partial \theta_j} J(\theta) \quad (\text{for } j=0,1,\dots,n)$$

Optimization Algorithms:

- Gradient descent
- Conjugant gradient
- BFGS
- L-BFGS

Advantages:

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

Disadvantages:

- More complex

EXAMPLE

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

Result: $\min_{\theta} J(\theta)$
 $\theta_1=5, \theta_2=5$

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

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

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

```
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);
```

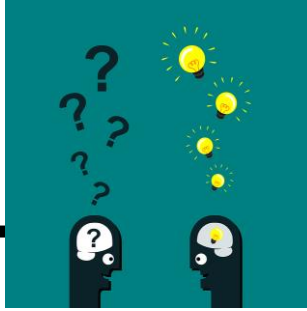
```
[optTheta, functionalVal, exitFlag]
=fminunc(@costFunction, initialTheta, options);
```

EXAMPLE – LOGISTIC REGRESSION

$$\text{theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \dots \\ \theta_n \end{bmatrix}$$

```
function[jVal, gradient] = costFunction(theta)
    jVal = [code to compute J(θ)];
    gradient(1) = [code to compute  $\frac{\partial}{\partial \theta_0} J(\theta)$ ];
    gradient(2) = [code to compute  $\frac{\partial}{\partial \theta_1} J(\theta)$ ];
    ...
    gradient(n+1) = [code to compute  $\frac{\partial}{\partial \theta_n} J(\theta)$ ];
```

QUESTION



Suppose you want to use an advanced optimization algorithm to minimize the cost function for logistic regression with parameters θ_0 and θ_1 . You write the following code:

```
function[jVal, gradient] = costFunction(theta)
jVal = % code to compute J(theta)
    gradient(1) = CODE#1 derivative for theta_0
    gradient(2) = CODE#2 derivative for theta_1
```

What should CODE#1 and CODE#2 above compute?

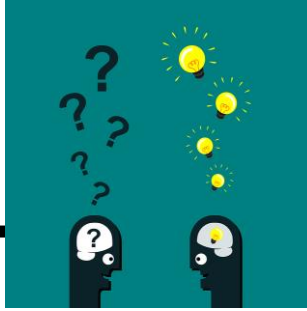
A: CODE#1 and CODE#2 should compute $J(\theta)$.

B: CODE#1 should be $\theta(1)$ and CODE#2 should be $\theta(2)$.

C: CODE#1 should compute $\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$ and CODE#2 should compute $\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_1$

D: None of the above.

QUESTION



Suppose you want to use an advanced optimization algorithm to minimize the cost function for logistic regression with parameters θ_0 and θ_1 . You write the following code:

```
function[jVal, gradient] = costFunction(theta)
jVal = % code to compute J(theta)
    gradient(1) = CODE#1 derivative for theta_0
    gradient(2) = CODE#2 derivative for theta_1
```

What should CODE#1 and CODE#2 above compute?

A: CODE#1 and CODE#2 should compute $J(\theta)$.

B: CODE#1 should be $\theta(1)$ and CODE#2 should be $\theta(2)$.

~~C: CODE#1 should compute $\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_0$ and CODE#2 should compute $\frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_1$~~

D: None of the above.

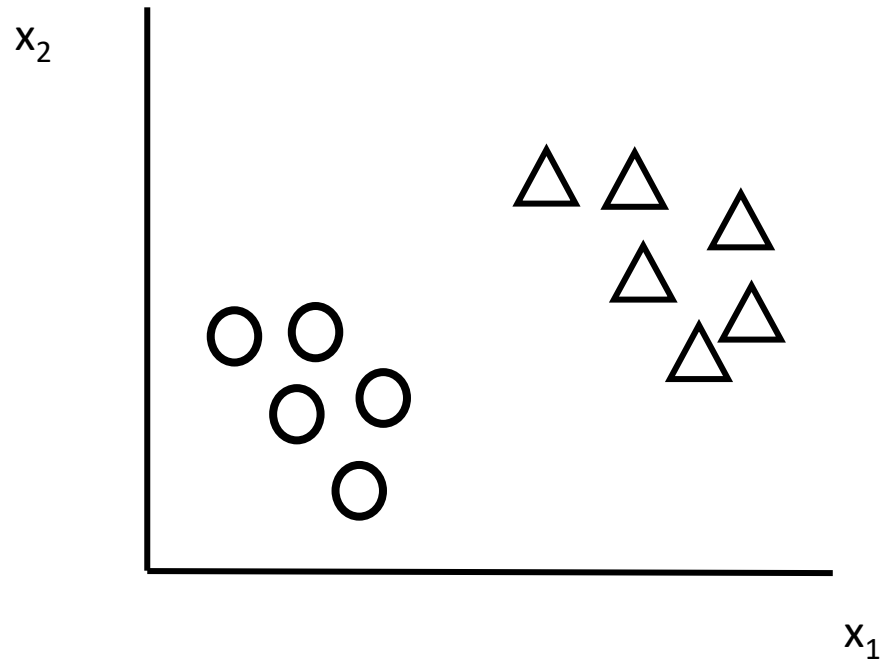
MULTICLASS CLASSIFICATION

- Email foldering/tagging: work, friends, family, hobby
- Medical diagrams: Not ill, cold, flu
- Weather: sunny, cloudy, rainy, snow

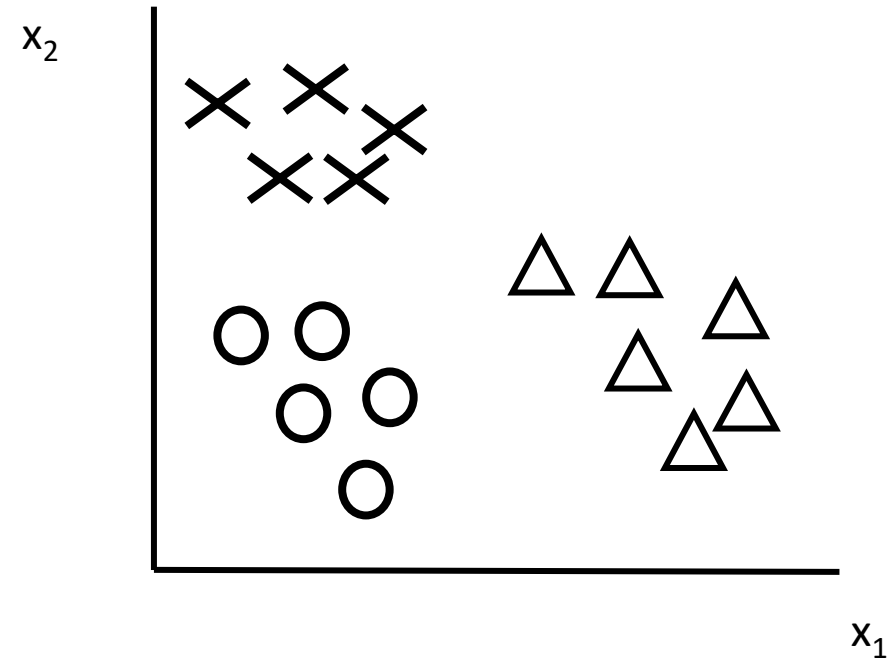
→ $y \in \{0, 1, \dots, n\}$

MULTICLASS CLASSIFICATION

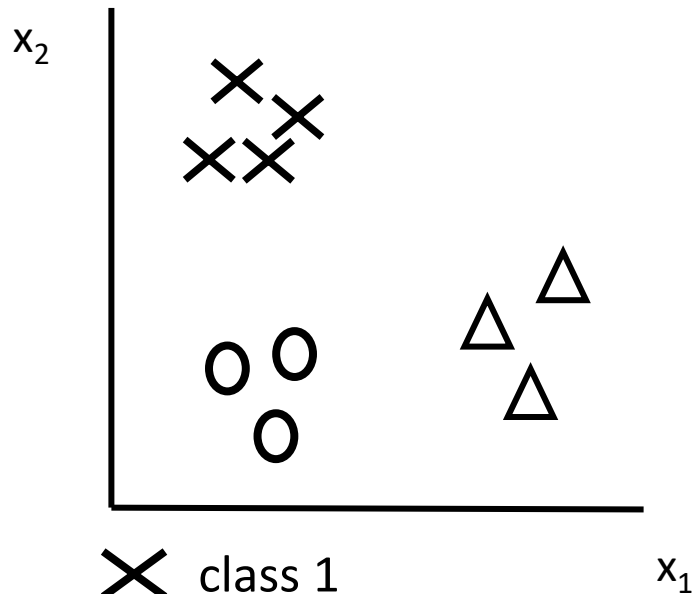
Binary



Multiclass



ONE-VS-ALL (ONE-VS-REST)

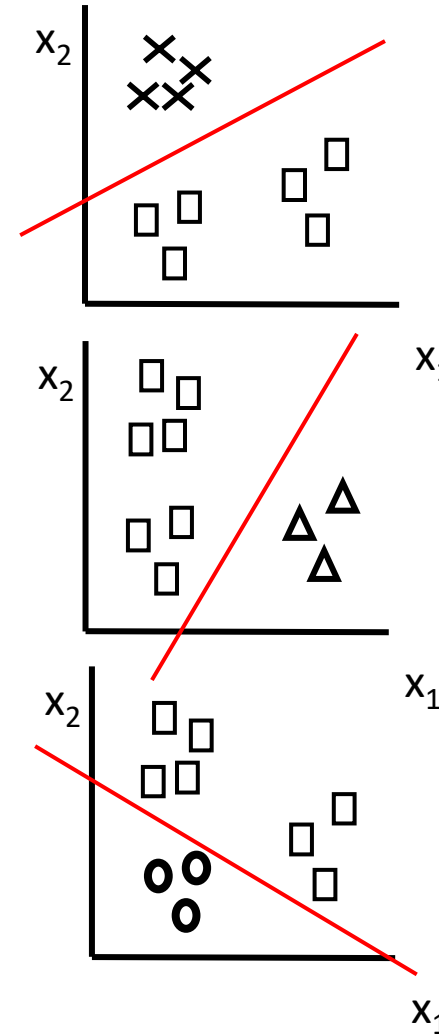


× class 1
△ class 2
○ class 3

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

→ Three classifiers – each trained to recognize one of the three classes

New training sets



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

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

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

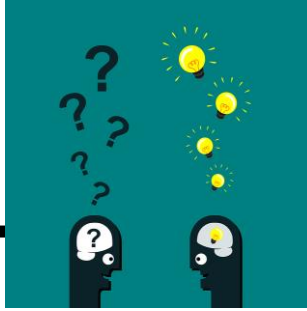
$$h_{\theta}^{(3)}(x)$$

ONE-VS-ALL (ONE-VS-REST)

- Train a logistic regression classifier $h_{\theta}^{(i)}(x)$ for each class i to predict the probability that $y = i$.
- To make a prediction on new input x , pick the class i that maximizes

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

QUESTION



Suppose you have a multi-class classification problem with k classes (so $y \in \{1, 2, \dots, k\}$). Using the 1-vs.-all method, how many different logistic regression classifiers will you end up training?

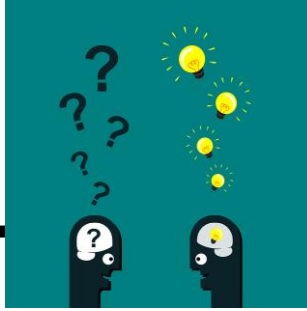
A: $k-1$

B: k

C: $k+1$

D: Approximately $\log_2(k)$

QUESTION



Suppose you have a multi-class classification problem with k classes (so $y \in \{1, 2, \dots, k\}$). Using the 1-vs.-all method, how many different logistic regression classifiers will you end up training?

A: $k-1$

~~B: k~~

C: $k+1$

D: Approximately $\log_2(k)$

WRAP-UP

Classification

- To attempt classification, one method is to use linear regression and map all predictions greater than 0.5 as a 1 and all less than 0.5 as a 0. However, this method does not work well because classification is not actually a linear function.
- The classification problem is just like the regression problem, except that the values we now want to predict take on only a small number of discrete values. For now, we will focus on the binary classification problem in which y can take on only two values, 0 and 1. For instance, if we are trying to build a spam classifier for email, then $x^{(i)}$ may be some features of a piece of email, and y may be 1 if it is a piece of spam mail, and 0 otherwise. Hence, $y \in \{0,1\}$. 0 is also called the negative class, and 1 the positive class, and they are sometimes also denoted by the symbols “-” and “+.” Given $x^{(i)}$, the corresponding $y^{(i)}$ is also called the label for the training example.

WRAP-UP

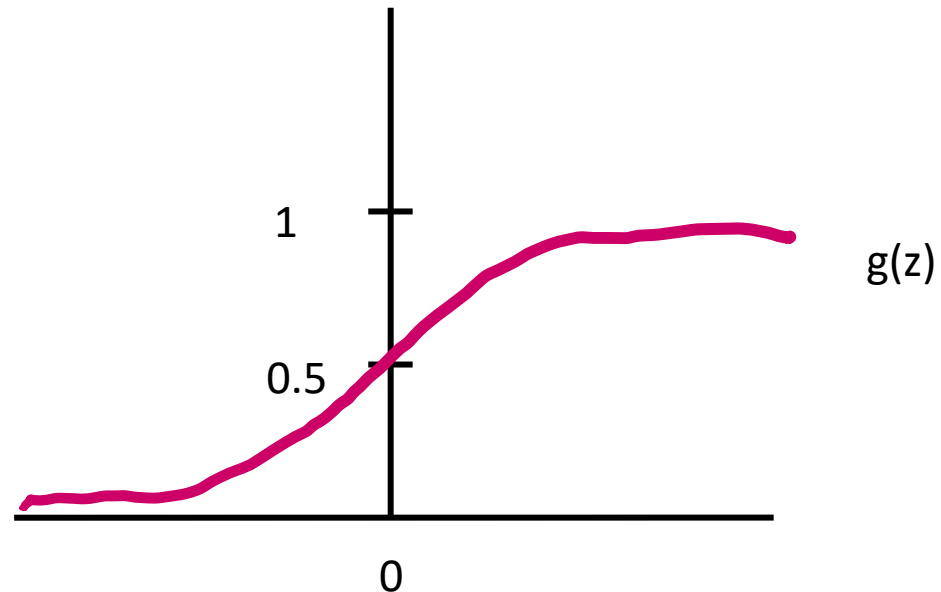
Hypothesis Representation

- We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given x . However, it is easy to construct examples where this method performs very poorly. Intuitively, it also does not make sense for $h_{\theta}(x)$ to take values larger than 1 or smaller than 0 when we know that $y \in \{0, 1\}$. To fix this, let us change the form for our hypotheses $h_{\theta}(x)$ to satisfy $0 \leq h_{\theta}(x) \leq 1$. This is accomplished by plugging $\theta^T x$ into the Logistic Function.

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

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

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

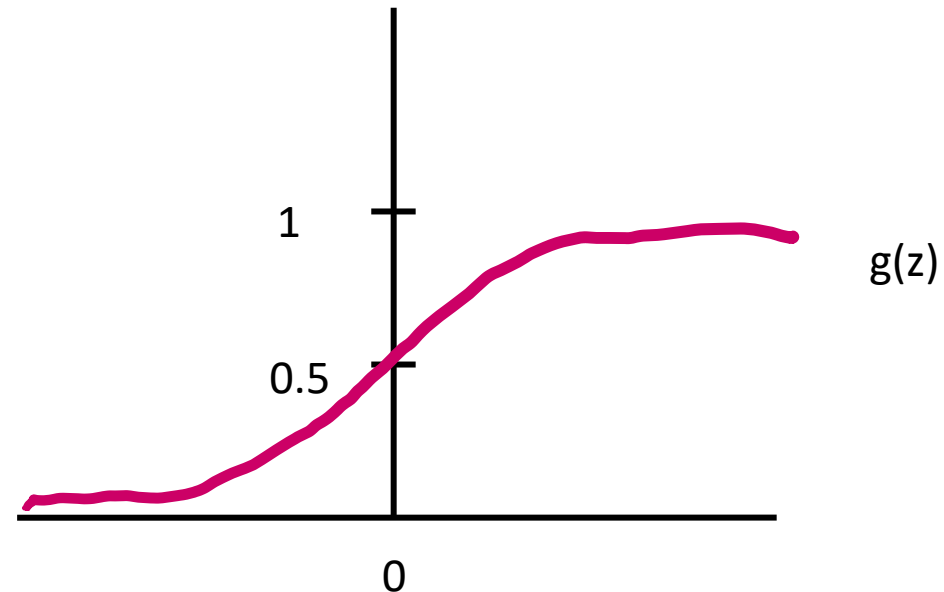


WRAP-UP

- The function $g(z)$, shown here, maps any real number to the $(0, 1)$ interval, making it useful for transforming an arbitrary-valued function into a function better suited for classification.
- $h_{\theta}(x)$ will give us the probability that our output is 1. Our probability that our prediction is 0 is just the complement of our probability that it is 1.

$$\rightarrow p(y=1|x; \theta) + p(y=0|x; \theta) = 1$$

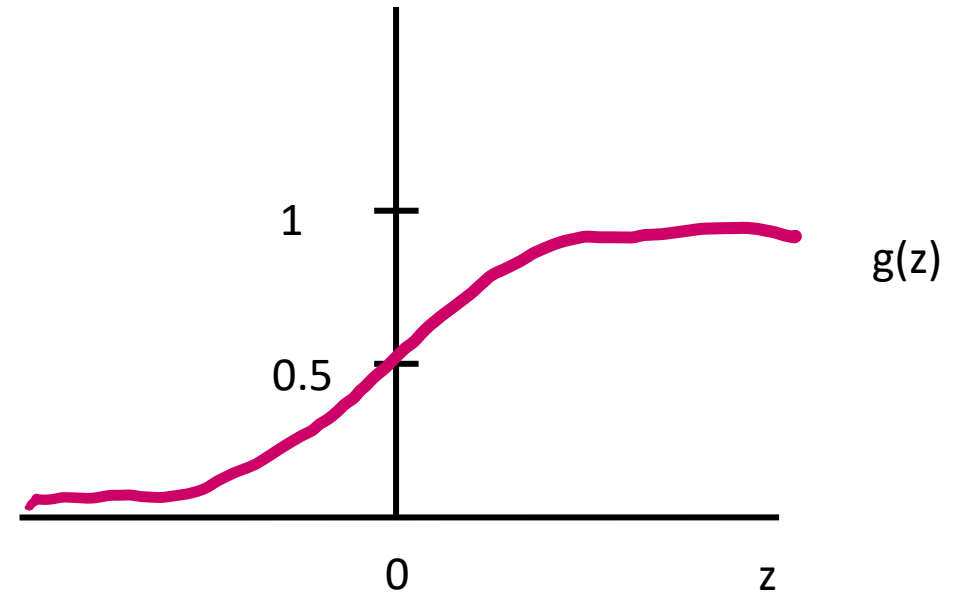
$$\rightarrow p(y=0|x; \theta) = 1 - p(y=1|x; \theta)$$



WRAP-UP

Decision Boundary

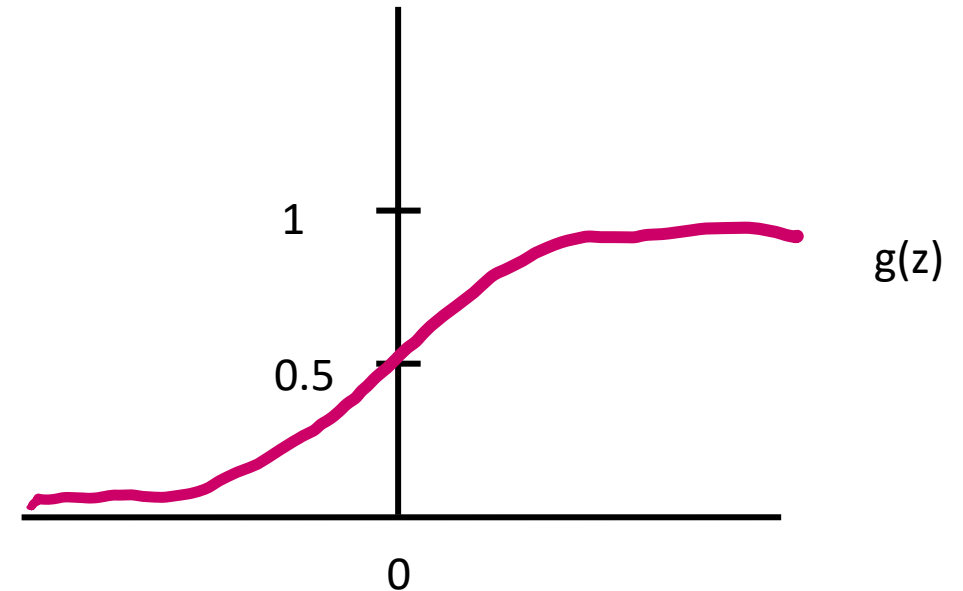
- In order to get our discrete 0 or 1 classification, we can translate the output of the hypothesis function as follows:
- $h_{\theta}(x) \geq 0.5 \rightarrow y=1$
- $h_{\theta}(x) < 0.5 \rightarrow y=0$
- The way our logistic function g behaves is that when its input is greater than or equal to zero, its output is greater than or equal to 0.5:
- $g(z) \geq 0.5$ when $z \geq 0$
- So if our input to g is $\theta^T X$ then that means:
- $h_{\theta}(x) = g(\theta^T X) \geq 0.5$
- when $\theta^T X \geq 0$



WRAP-UP

Decision Boundary

- So if our input to g is $\theta^T X$ then that means:
- $h_{\theta}(x) = g(\theta^T X) \geq 0.5$
- when $\theta^T X \geq 0$
- From these statements we can now say:
- $\theta^T X \geq 0 \Rightarrow y=1$
- $\theta^T X < 0 \Rightarrow y=0$
- The decision boundary is the line that separates the area where $y = 0$ and where $y = 1$. It is created by our hypothesis function.
- The input to the sigmoid function $g(z)$ does not need to be linear and could be a function that describes a circle (e. g.: $z = \theta_0 + \theta_1(x_1)^2 + \theta_2(x_2)^2$) or any shape to fit our data.



WRAP-UP

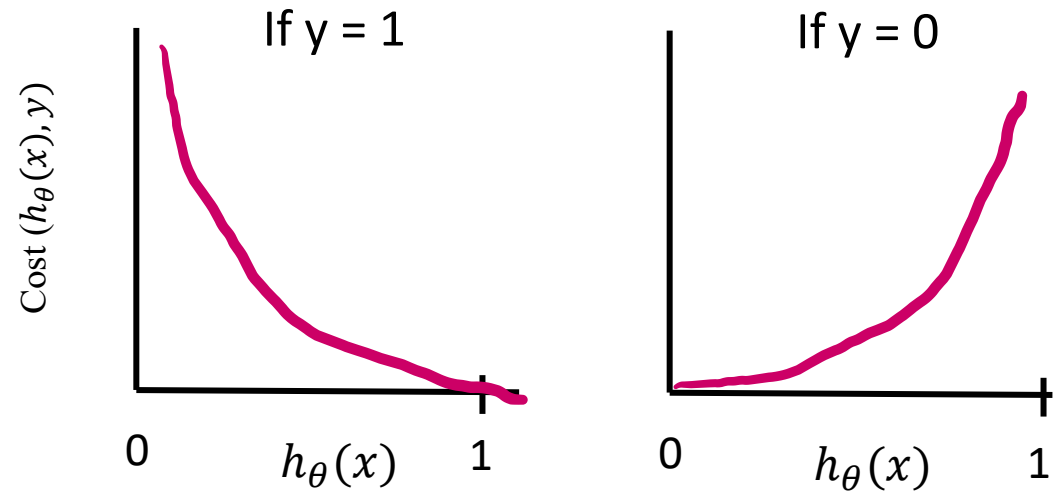
Cost Function Logistic Regression

- We cannot use the same cost function that we use for linear regression because the Logistic Function will cause the output to be wavy, causing many local optima. In other words, it will not be a convex function.
- Instead, our cost function for logistic regression looks like:

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

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

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



WRAP-UP

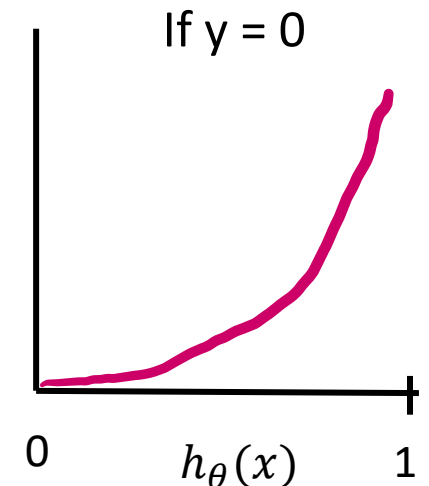
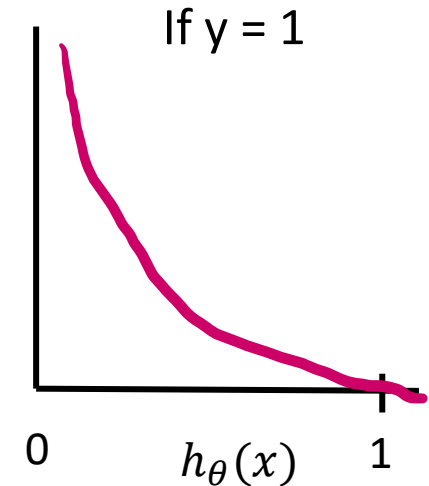
Cost Function Logistic Regression

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

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

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

- If our correct answer 'y' is 0, then the cost function will be 0 if our hypothesis function also outputs 0. If our hypothesis approaches 1, then the cost function will approach infinity.
- If our correct answer 'y' is 1, then the cost function will be 0 if our hypothesis function outputs 1. If our hypothesis approaches 0, then the cost function will approach infinity.
- Note that writing the cost function in this way guarantees that $J(\theta)$ is convex for logistic regression.



WRAP-UP

We can compress our cost function's two conditional cases into one case:

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

Notice that when y is equal to 1, then the second term $(1-y) \log(1 - h_{\theta}(x))$ will be zero and will not affect the result. If y is equal to 0, then the first term $-y \log(h_{\theta}(x))$ will be zero and will not affect the result.

We can fully write out our entire cost function as follows:

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

A vectorized implementation is:

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot -y^T \log(h) - (1-y)^T \log(1-h)$$

WRAP-UP

Simplified Cost Function and Gradient Descent

Remember that the general form of gradient descent is:

Repeat {

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

(simultaneously update all θ_j)

}

We can work out the derivative part using calculus to get:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j$$

(simultaneously update all θ_j)

}

A vectorized implementation is:

$$\theta := \theta - \alpha \frac{1}{m} X^T (g(X\theta) - \vec{y})$$

WRAP-UP

Advanced Optimization

"Conjugate gradient", "BFGS", and "L-BFGS" are more sophisticated, faster ways to optimize θ that can be used instead of gradient descent.

In order use libraries that implement these algorithms, we can provide a function that evaluates the following two functions for a given input value θ :

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

We can write a single function that returns both of these:

```
function [jVal, gradient] = costFunction(theta)
    jVal = [...code to compute J(theta)...];
    gradient = [...code to compute derivative of J(theta)...];
end
```

WRAP-UP

Advanced Optimization

Then we can use e.g. "fminunc()" optimization algorithm along with the "optimset()" function that creates an object containing the options we want to send to "fminunc()".

```
options = optimset('GradObj', 'on', 'MaxIter', 100);  
initialTheta = zeros(2,1);  
[optTheta, functionVal, exitFlag] = fminunc(@costFunction, initialTheta,  
options);
```

We give to the function "fminunc()" our cost function, our initial vector of theta values, and the "options" object that we created beforehand.

WRAP-UP

Multiclass Classification: One-vs-all

Now we will approach the classification of data when we have more than two categories. Instead of $y = \{0,1\}$ we will expand our definition so that $y = \{0,1,...,n\}$.

Since $y = \{0,1,...,n\}$, we divide our problem into $n+1$ binary classification problems; in each one, we predict the probability that 'y' is a member of one of our classes.

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

$$h_{\theta}^{(i)}(x) = p(y = i|x; \theta) \quad (i = 0,...,n)$$

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

We are basically choosing one class and then putting all the others into a single second class. We do this repeatedly, applying binary logistic regression to each case, and then use the hypothesis that returned the highest value as our prediction.

To summarize:

- Train a logistic regression classifier $h_{\theta}(x)$ for each class to predict the probability that $y = i$.
- To make a prediction on a new x , pick the class that maximizes $h_{\theta}(x)$.

QUIZ – QUESTION 1

Suppose that you have trained a logistic regression classifier, and it outputs on a new example x a prediction $h_{\theta}(x) = 0.2$. This means (check all that apply):

A: Our estimate for $P(y=0|x;\theta)$ is 0.8.

B: Our estimate for $P(y=0|x;\theta)$ is 0.2.

C: Our estimate for $P(y=1|x;\theta)$ is 0.8.

D: Our estimate for $P(y=1|x;\theta)$ is 0.2.

QUIZ – QUESTION 1

Suppose that you have trained a logistic regression classifier, and it outputs on a new example x a prediction $h_{\theta}(x) = 0.2$. This means (check all that apply):

~~A~~: Our estimate for $P(y=0|x;\theta)$ is 0.8.

B: Our estimate for $P(y=0|x;\theta)$ is 0.2.

C: Our estimate for $P(y=1|x;\theta)$ is 0.8.

~~D~~: Our estimate for $P(y=1|x;\theta)$ is 0.2.

QUIZ – QUESTION 2

Suppose you have the following training set, and fit a logistic regression classifier $h_{\theta}(x)=g(\theta_0+\theta_1x_1+\theta_2x_2)$. Which of the following are true? Check all that apply.

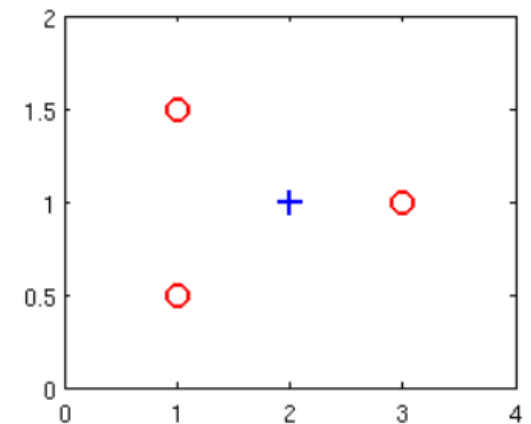
A: Adding polynomial features could increase how well we can fit the training data.

B: At the optimal value of θ (e.g., found by fminunc), we will have $J(\theta) \geq 0$.

C: Adding polynomial features would increase $J(\theta)$ because we are now summing over more terms.

D: If we train gradient descent for enough iterations, for some examples $x^{(i)}$ in the training set it is possible to obtain $h_{\theta}(x^{(i)}) > 1$.

x_1	x_2	y
1	0.5	0
1	1.5	0
2	1	1
3	1	0



QUIZ – QUESTION 2

Suppose you have the following training set, and fit a logistic regression classifier $h_{\theta}(x)=g(\theta_0+\theta_1x_1+\theta_2x_2)$. Which of the following are true? Check all that apply.

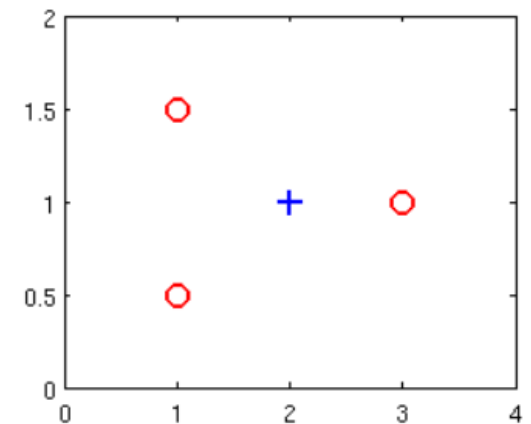
☒ A: Adding polynomial features could increase how well we can fit the training data.

☒ B: At the optimal value of θ (e.g., found by fminunc), we will have $J(\theta) \geq 0$.

☐ C: Adding polynomial features would increase $J(\theta)$ because we are now summing over more terms.

☐ D: If we train gradient descent for enough iterations, for some examples $x^{(i)}$ in the training set it is possible to obtain $h_{\theta}(x^{(i)}) > 1$.

x_1	x_2	y
1	0.5	0
1	1.5	0
2	1	1
3	1	0



QUIZ – QUESTION 3

For logistic regression, the gradient is given by $\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j$

Which of these is a correct gradient descent update for logistic regression with a learning rate of α ? Check all that apply.

A: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m \left(\frac{1}{1+e^{-\theta^T x}} - y^{(i)} \right) * x^{(i)}_j$ (simultaneously update for all j).

B: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j]$ (simultaneously update for all j).

C: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}]$ (simultaneously update for all j).

D: $\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^m [\theta^T x - y^{(i)}] * x^{(i)}$

QUIZ – QUESTION 3

For logistic regression, the gradient is given by $\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j$

Which of these is a correct gradient descent update for logistic regression with a learning rate of α ? Check all that apply.

☒ A: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m \left(\frac{1}{1+e^{-\theta^T x}} - y^{(i)} \right) * x^{(i)}_j$ (simultaneously update for all j).

☒ B: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}_j]$ (simultaneously update for all j).

☐ C: $\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m [(h_{\theta}(x^{(i)}) - y^{(i)}) * x^{(i)}]$ (simultaneously update for all j).

☐ D: $\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^m [\theta^T x - y^{(i)}] * x^{(i)}$

QUIZ – QUESTION 4

Which of the following statements are true? Check all that apply.

A: Linear regression always works well for classification if you classify by using a threshold on the prediction made by linear regression.

B: The sigmoid function $g(z) = \frac{1}{1+e^{-z}}$ is never greater than one (>1).

C: The cost function $J(\theta)$ for logistic regression trained with $m \geq 1$ examples is always greater than or equal to zero.

D: For logistic regression, sometimes gradient descent will converge to a local minimum (and fail to find the global minimum). This is the reason we prefer more advanced optimization algorithms such as fminunc (conjugate gradient/BFGS/L-BFGS/etc.).

QUIZ – QUESTION 4

Which of the following statements are true? Check all that apply.

A: Linear regression always works well for classification if you classify by using a threshold on the prediction made by linear regression.

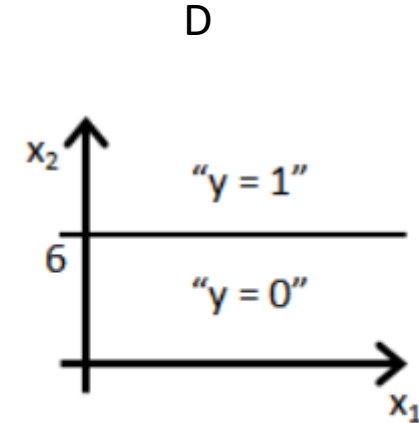
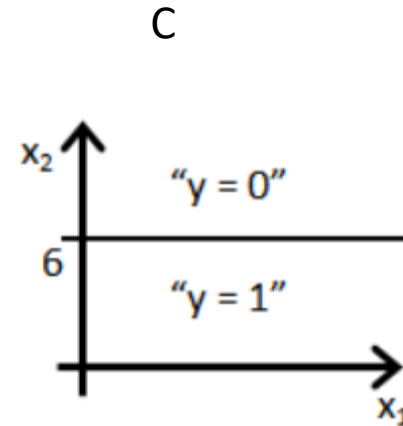
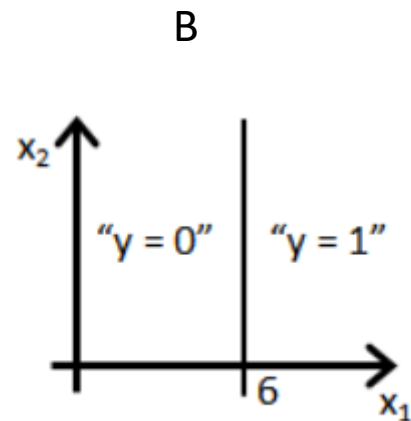
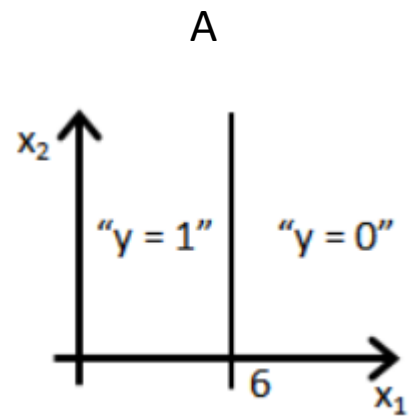
~~B:~~ The sigmoid function $g(z) = \frac{1}{1+e^{-z}}$ is never greater than one (>1).

~~C:~~ The cost function $J(\theta)$ for logistic regression trained with $m \geq 1$ examples is always greater than or equal to zero.

D: For logistic regression, sometimes gradient descent will converge to a local minimum (and fail to find the global minimum). This is the reason we prefer more advanced optimization algorithms such as fminunc (conjugate gradient/BFGS/L-BFGS/etc.).

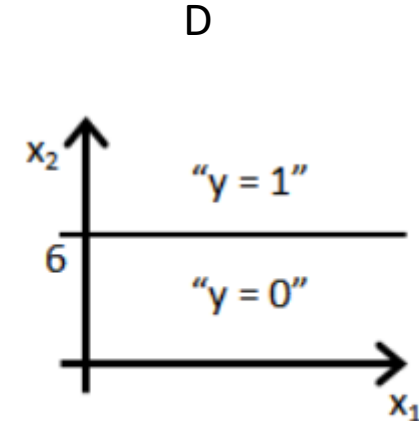
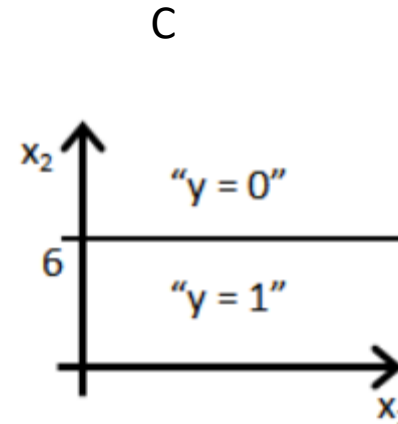
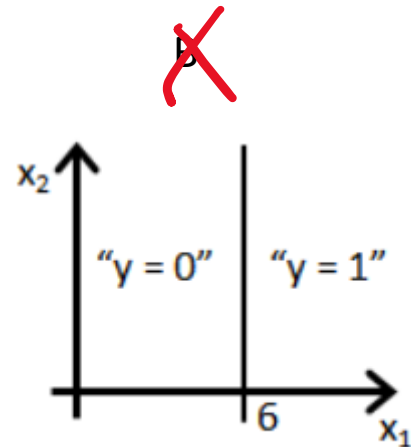
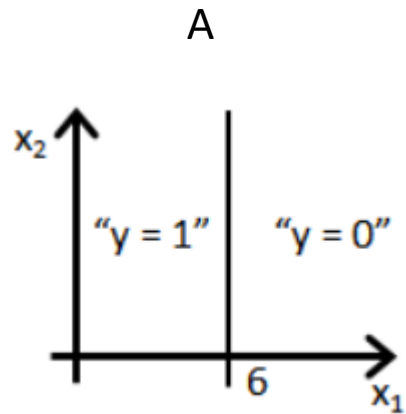
QUIZ - QUESTION 5

- Suppose you train a logistic classifier $h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$. Suppose $\theta_0 = -6$, $\theta_1 = 1$, $\theta_2 = 0$. Which of the following figures represents the decision boundary found by your classifier?



QUIZ - QUESTION 5

- Suppose you train a logistic classifier $h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$. Suppose $\theta_0 = -6$, $\theta_1 = 1$, $\theta_2 = 0$. Which of the following figures represents the decision boundary found by your classifier?



$$y = 1 \text{ if } -6 + x_1 \geq 0 \rightarrow \text{If } x_1 \geq 6$$