CENG 463
Machine Learning

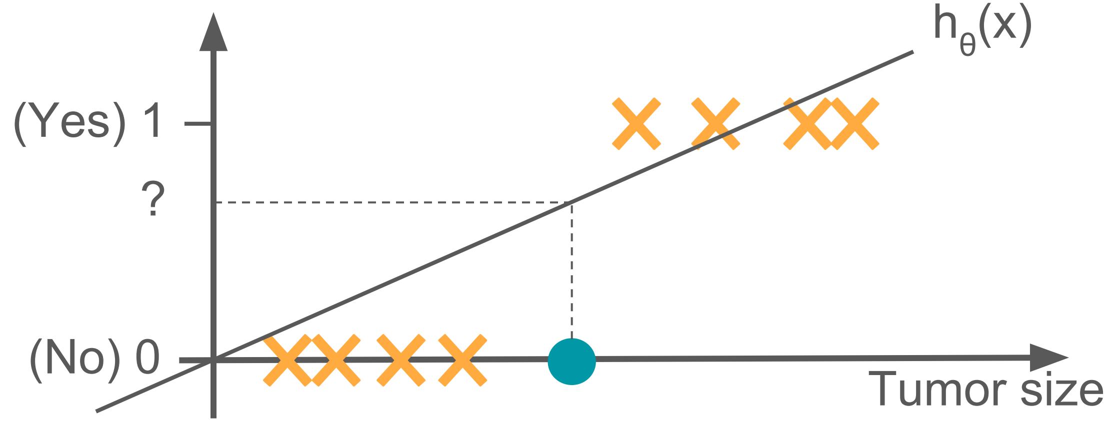
Lecture 06 - Logistic Regression

Slides were prepared using the material provided in Stanford's Machine Learning Course by Andrew Ng.

Classification

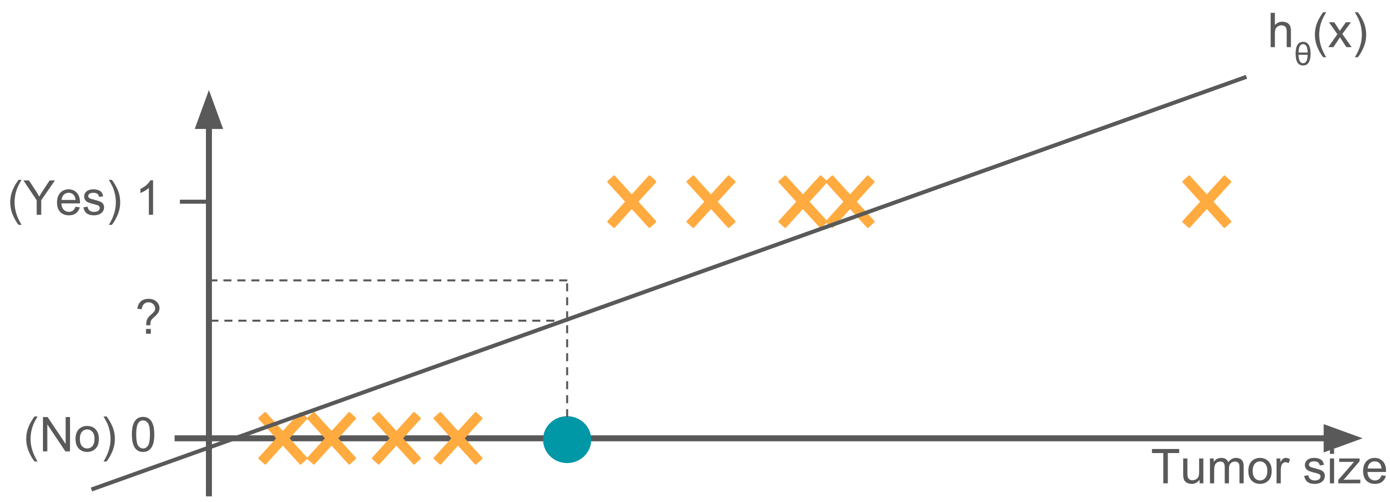
- Logistic Regression is a **classification method**!
- Examples:
  - Email: Spam / Not Spam?
  - Brain tumor: Malignant / Benign?
- $y \in \{0,1\}$
  - 0: Negative class (e.g. benign tumor)
  - 1: Positive class (e.g. malignant tumor)
- $y \in \{0,1,2,..\}$ if there are more than two classes.

Why not using Linear Regression?



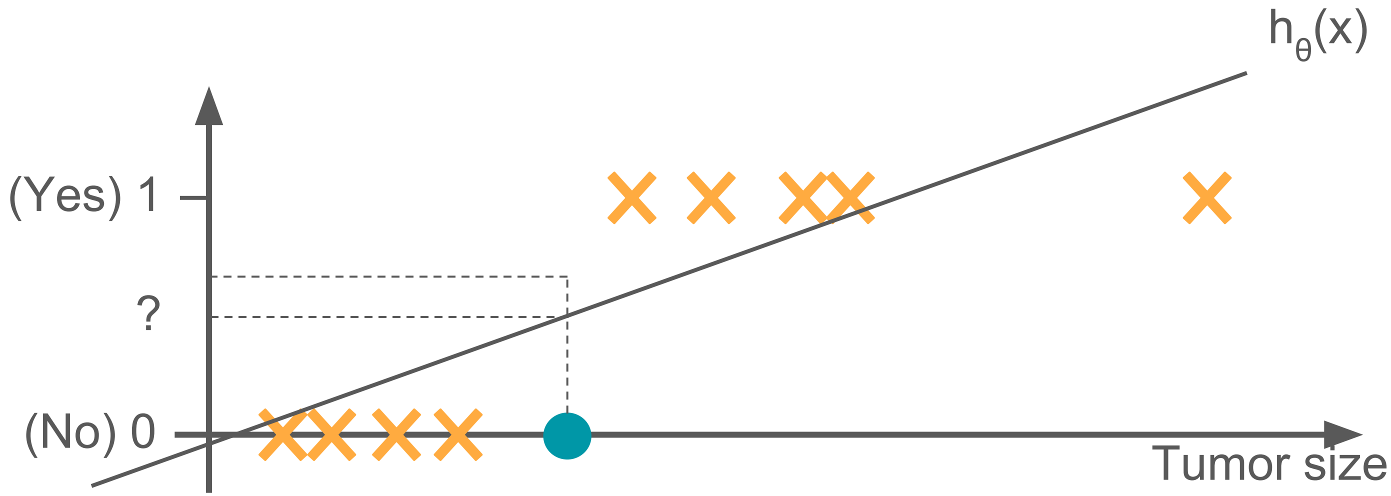
- Threshold classifier output $h_\theta(x)$ at 0.5:
  - If $h_\theta(x)>0.5$, predict 'malignant', otherwise predict 'benign'.

Why not using Linear Regression?



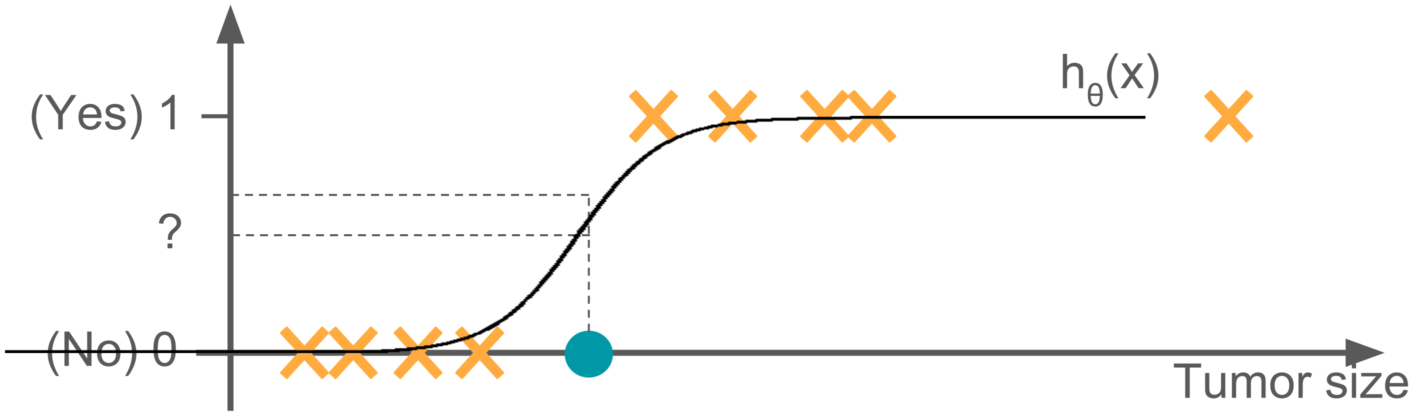
- But, what if we have an extreme case sample in the training set?
  - For the above scenario, linear regression is not suitable.
  - The best fitting line changed significantly because of a single very large tumor size.

Why not using Linear Regression?



- In fact, for most of the classification problems, linear regression, even polynomial regression is not suitable.
- We also want $h_\theta(x)$ to take values between 0 and 1. With linear regression $h_\theta(x)$ can take <0 and >1 values.
- One solution is: **Logistic Regression**, where $0 < h_\theta(x) < 1$

Logistic Regression



- In logistic regression, we model our hypothesis $h_\theta(x)$, so that it takes values between 0 and 1. **How?**

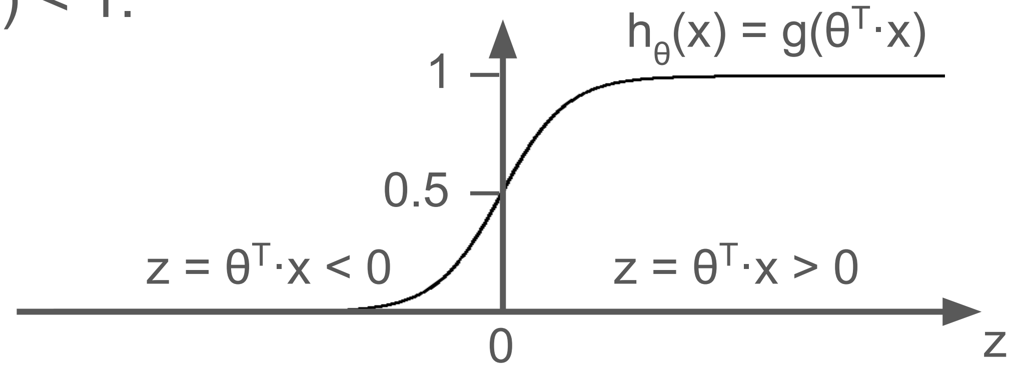
Logistic Regression

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

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

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

- **g(z)** is called **Logistic** or **Sigmoid** function.
- Logistic regression fits the parameters (θ) to this model.
- θ here are not the same with linear regression parameters.
- As we wanted, $0 < h_\theta(x) < 1$.



Logistic Regression

- We interpret the logistic regression output as follows:
  $h_\theta(x)$ = estimated probability that y = 1 on input x

- For tumor example, if $h_\theta(x) = 0.7$, we say that the patient has 70% chance to have a malignant tumor.
- In statistics notation:
  $h_\theta(x) = P(y=1 \mid x; \theta)$

- In words:
  "probability that y = 1, given x, parameterized by θ "

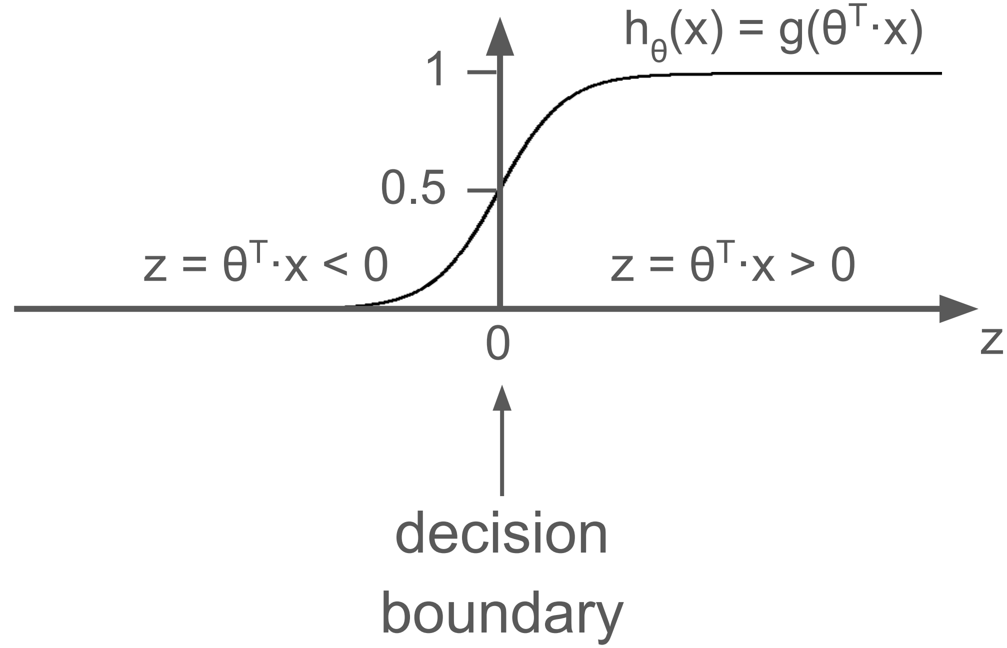
- Keep in mind that: $P(y=0 \mid x; \theta) + P(y=1 \mid x; \theta) = 1$

Decision Boundary

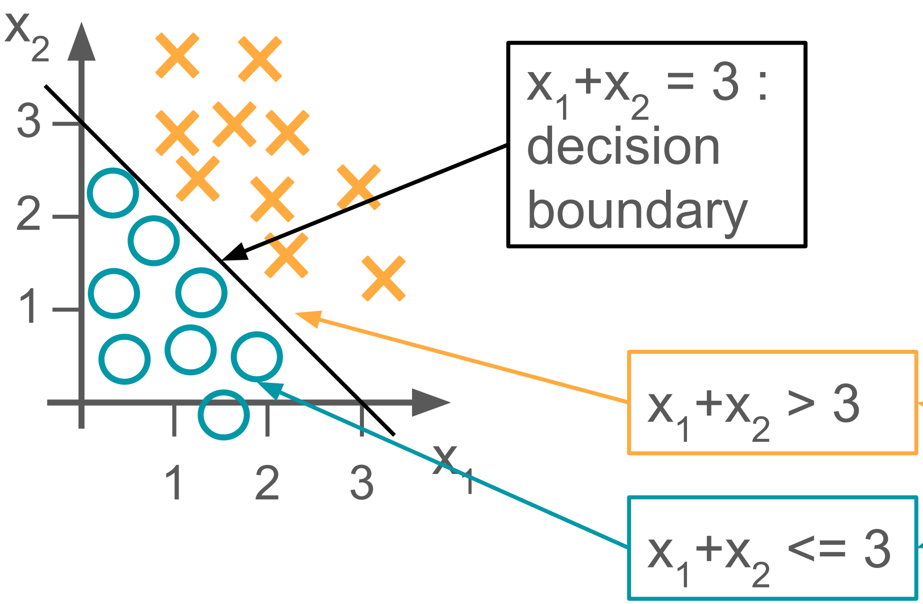
- One strategy for decision is to predict:
  - $y=1$ if $h_\theta(x) > 0.5$
  - $y=0$ if $h_\theta(x) <= 0.5$
- This means to predict:
  - $y=1$ if $\theta^T \cdot x > 0$
  - $y=0$ if $\theta^T \cdot x <= 0$

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

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

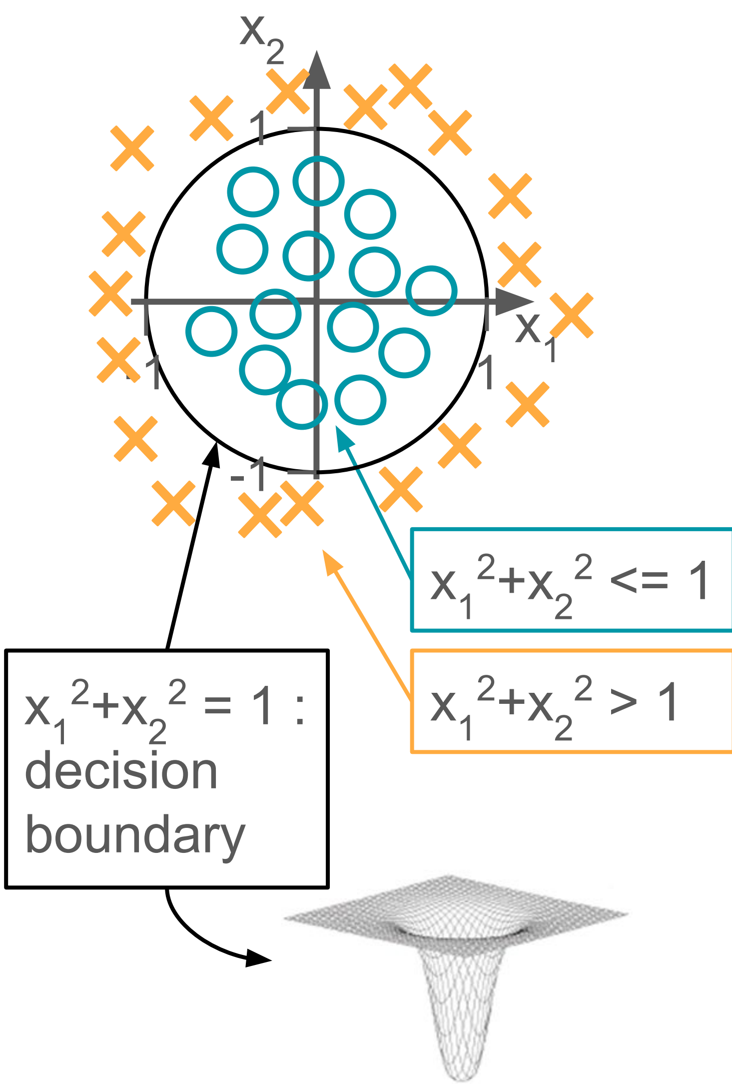


Decision Boundary



- $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$
- Let's say:
  - $\theta_0 = -3, \theta_1 = 1, \theta_2 = 1$
- Using the same strategy, we predict:
  - $y=1$ if $-3+x_1+x_2 > 0$
  - $y=0$ if $-3+x_1+x_2 <= 0$

Decision Boundary



- An example to non-linear decision boundary:
  $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$
- Let's say:
  - $\theta_0 = -1$, $\theta_1 = \theta_2 = 0$, $\theta_3 = \theta_4 = 1$
- Using the same strategy, we predict:
  - $y=1$ if $-1+x_1^2+x_2^2 > 0$
  - $y=0$ if $-1+x_1^2+x_2^2 <= 0$

Cost Function

- Now, we need to estimate parameters (θ) for the decision boundary.
- If we have m samples, n features, 2 classes:
  - $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})$
  - $x^{(i)} = [\ x_0^{(i)}\ \ x_1^{(i)}\ \ \dots\ \ x_n^{(i)}\ ]^T$
  - $y \in \{0,1\}$

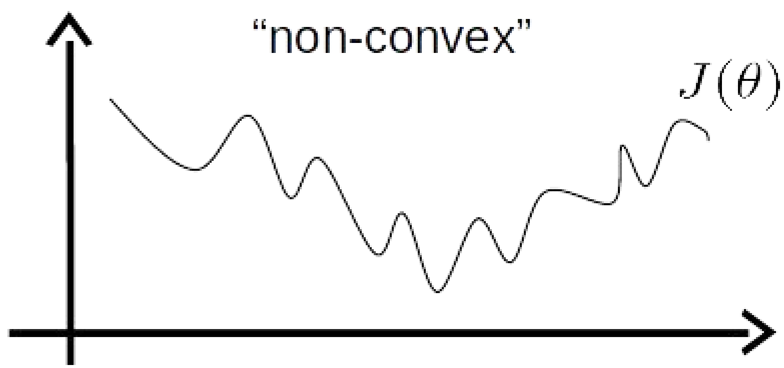
$$h_\theta(x) = \frac{1}{1+e^{-\theta^T \cdot x}} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ .. \\ y^{(m)} \end{bmatrix} \qquad X = \begin{bmatrix} .. & x^{(1)T} & .. \\ .. & x^{(2)T} & .. \\ .. & .. & .. \\ .. & x^{(m)T} & .. \end{bmatrix}$$

Cost Function

- Remember the cost function in linear regression:

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

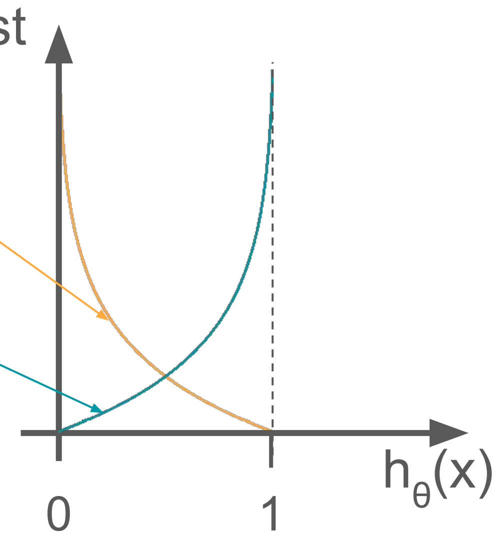
- It turns out that, because of the non-linearity of the sigmoid function that we use in $h_\theta(x)$, this cost function becomes non-convex (i.e. contains local minima).



Cost Function

- We need another function which is convex.
- **Logistic regression cost function:**

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



- For y = 1:
  - cost=0 if $h_\theta(x)=1$, cost $\rightarrow\infty$ as $h_\theta(x) \rightarrow 0$
- For y = 0:
  - cost=0 if $h_\theta(x)=0$, cost $\rightarrow\infty$ as $h_\theta(x) \rightarrow 1$
- If we predict 0 when y=1, or 1 when y = 0, we are penalized by a very large cost.

Cost Function

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

- Knowing that y is always equal to 0 or 1, we can define the cost function with a single line.

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

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

Gradient Descent

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

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

- We need to find the parameters that minimize θ.
- It turns out that, the derivative computed using calculus is <u>identical to the derivative term in linear regression</u>:

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

Gradient Descent

- Algorithm is also identical to linear regression:

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_1^{(i)}$$

...

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

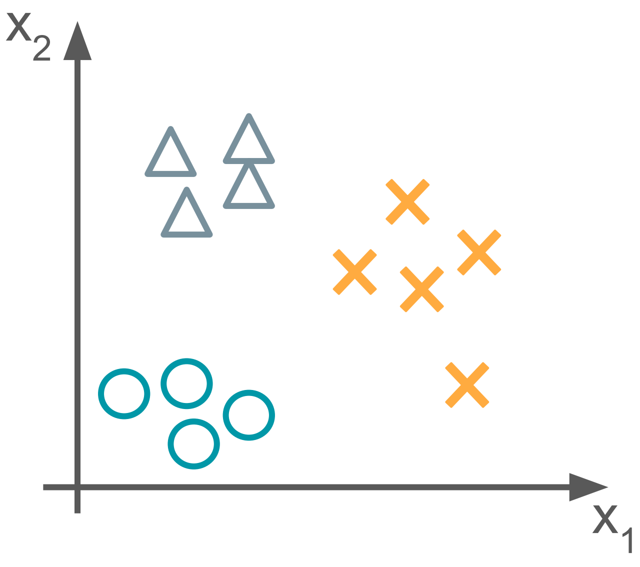
}

$x_0^{(i)} = 1$

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

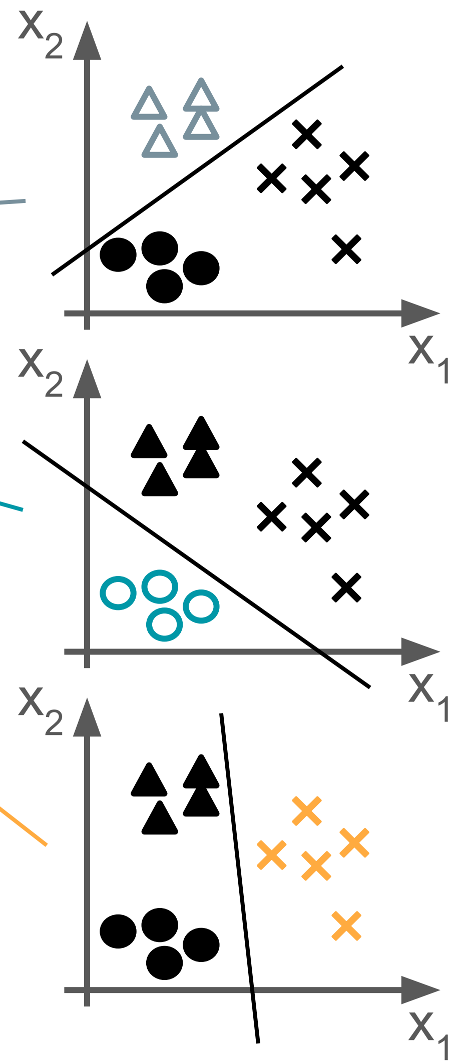
Multi-class Classification

- Examples:
  - Email tagging: Work, Friends, Family, Hobby
  - Weather: Sunny, Cloudy, Rain, Snow
  - Sports: Win the match, Lose the match, Draw
- Graphical representation for 3 classes and 2 features:



Multi-class Classification

- **One versus rest:**
  - Take y=1 for class 1 and y=0 for the rest. Train a two-class classifier: $h_\theta^{(1)}(x)$
  - Take y=1 for class 2 and y=0 for the rest. Train a two-class classifier: $h_\theta^{(2)}(x)$
  - Take y=1 for class 3 and y=0 for the rest. Train a two-class classifier: $h_\theta^{(3)}(x)$



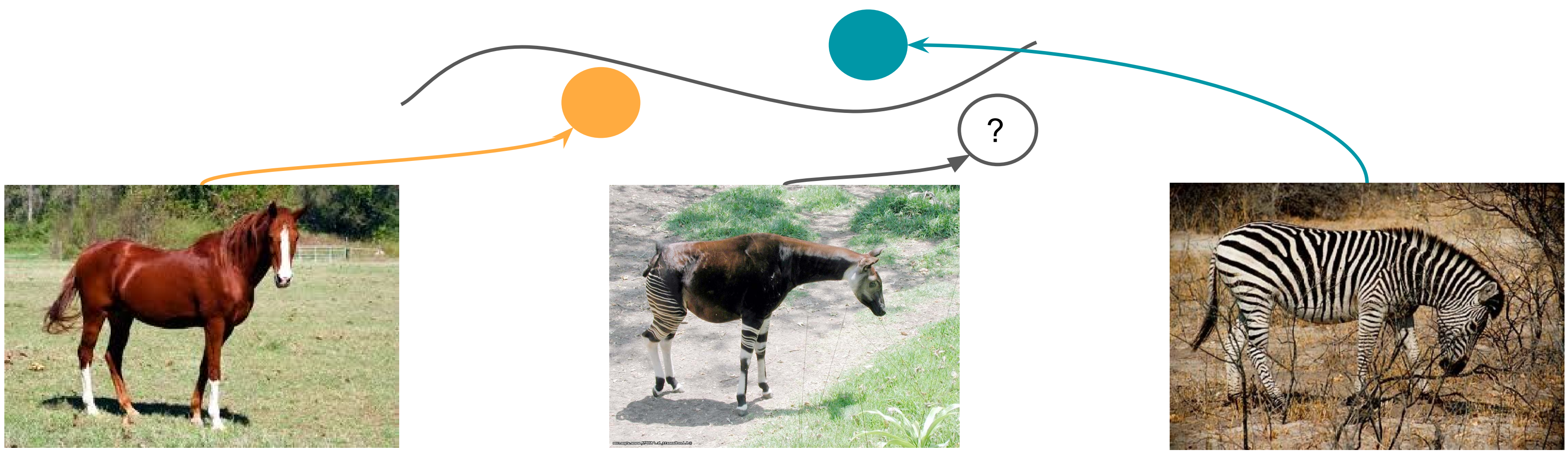
$h_\theta^{(i)}(x) = P(y=i \mid x; \theta)$ where i=1,2,3

Generative vs. Discriminative

- There are two main methods for learning algorithms:
  - Generative
  - Discriminative

Discriminative Methods

- Logistic regression is an example to a learning algorithm that models p(y|x), i.e. the conditional (posterior) probability of y given x. It is not interested in modeling classes y=1 or y=0.
- This kind of learning algorithms are called **discriminative**.
- Example: Direct modeling of p(zebra|image) and p(horse|image) based on some features of an animal.

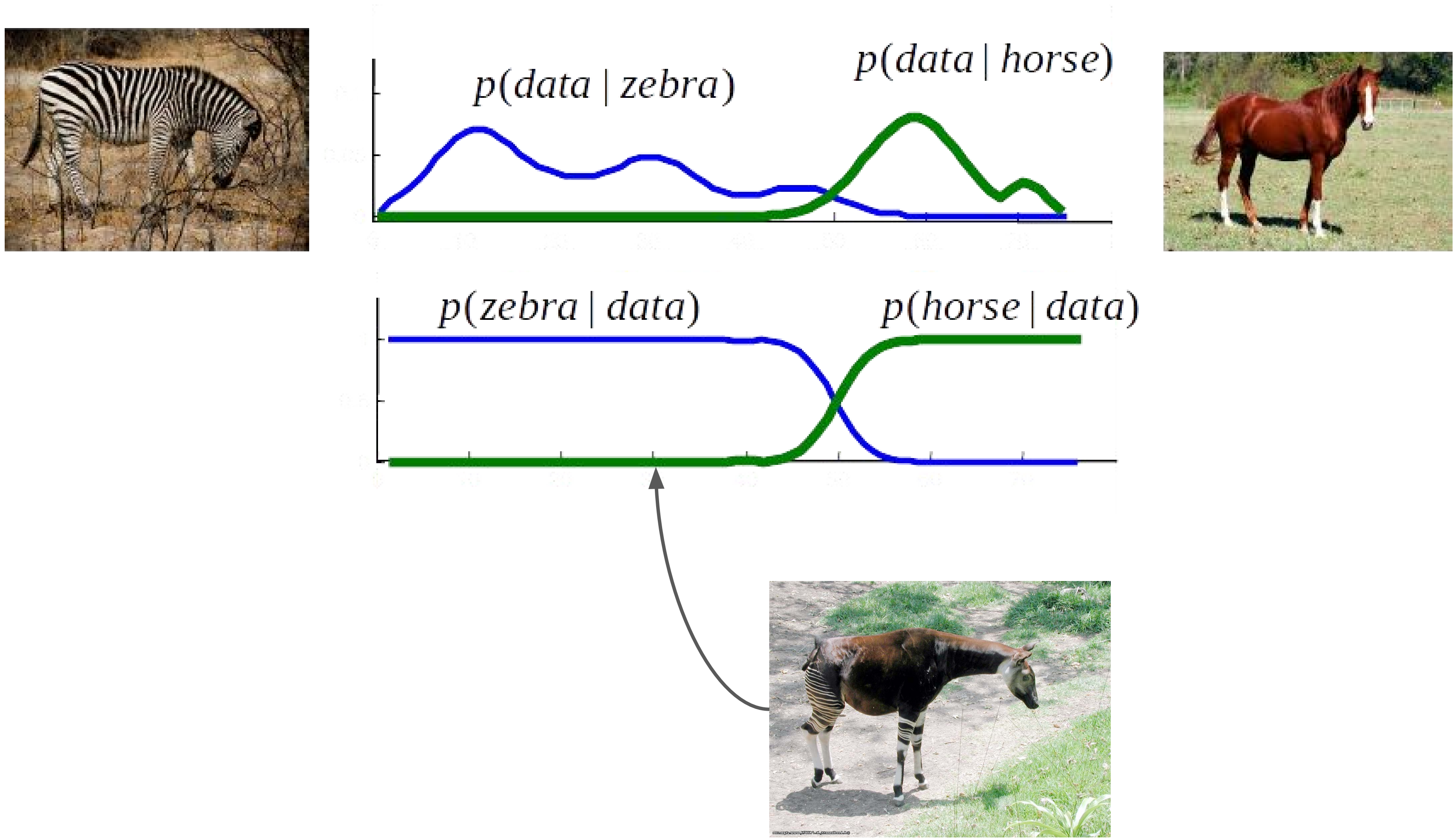


Generative Methods

- A different approach is first looking at samples of classes, and building separate models for them (e.g. building different models for zebra and horse).
- To classify a new animal, we compare it with the built models.
  - e.g. matching a new animal with zebra model and horse model, to see if it looks more like zebras or more like horses
- Such methods are called **generative**.
- That is what we did for Gaussian discriminant functions. We first built likelihoods, then obtained posteriors using the Bayes' rule.

$$p(zebra \mid image) = \frac{p(image \mid zebra) \cdot p(zebra)}{p(image)}$$

Generative Methods



Generative vs. Discriminative

- None of these methods can be called ‘the best’.
- Performance of these approaches depends on the problem.
- In general, if the distribution assumptions for the data (e.g. Gaussian) are correct, generative methods are expected to be better.
- On the other hand, if we are not sure about the underlying functions of the distributions, discriminative methods create more robust classifiers.

Summary

- We have learned about:
  - Logistic Regression
  - Decision Boundaries
  - Logistic Regression Cost Function
  - Multi-class Classification with Logistic Regression
  - Generative and Discriminative Methods