Note on Probability

$P \in [0,1]$

"probabilities lie in the range 0 to 1"

Probability of a random variable

$$P(Y = y) = P(y)$$

"The probability that random variable Y takes on value y"

Sum of probabilities

$$P(y = 0) + P(y = 1) = 1$$

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

"binary value"

"the sum of probabilities over all possible events is 1"

Conditional probability

$$P(y = 1 \mid x_1, x_2)$$

"The probability that $y = 1$, given x_1 and x_2"

Joint probability

$$P(y = 1, x = 2)$$

"The probability that $y = 1$ and $\mathrm{x} = 2$"

Note on Logarithms

Exponential ("to the power of"): $y = b^x$ (Exponent, base)

Logarithm is its inverse: $\log_b(y) = log_b(b^x) = x$

Natural logarithm: $\ln(x) = \log_e(x)$

(This uses Euler's constant as a base, $e \approx 2.71828$)

Logarithms cheat sheet

$$\log_b(b) = 1$$
$$\log_b(1) = 0$$

$$\log_b(p \cdot q) = \log_b(p) + \log_b(q)$$
$$\log_b(p/q) = \log_b(p) - \log_b(q)$$

$$\log_b(x^q) = q \log_b(x)$$

$$\lim_{x \to 0} \log_b(x) \to -\infty$$
$$\lim_{x \to \infty} \log_b(x) \to +\infty$$

Logistic Regression

Dr Jamie A Ward

Lecture 6

Lecture 6: Logistic Regression

Logistic regression

- Regression and classification
- The sigmoid (logistic) function
- Optimisation for logistic regression

Further issues

- Multi-class classification
- Overfitting and Underfitting
- Regularisation

Supervised Classification

Email

- ►Spam / not spam (binary classification)
- ►Personal / Social / Advertising (multi-class classification)

Health

- ►Disease yes / no (binary)

How did you get here

- ►Train / car / walk / bike (multi-class)

Binary classification:

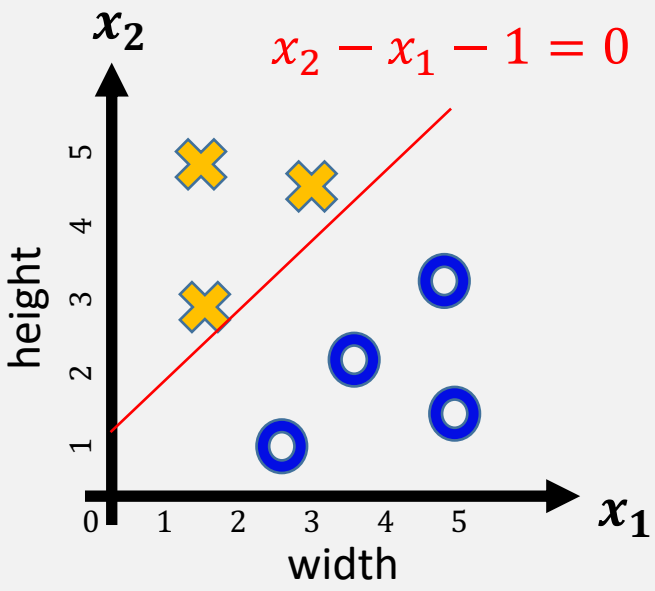
target / output / label

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

0: “negative” class 1: “positive” class

Linear classification: revision

Linear classification is about using a line to separate data into different classes



Question: Given a point $(x_1, x_2) = (4,3)$, what would this classifier predict for that point?

If $x_2 - x_1 - 1 = 0 \rightarrow$ lies on the line
If $x_2 - x_1 - 1 > 0 \rightarrow$ above the line (class ✖)
If $x_2 - x_1 - 1 < 0 \rightarrow$ below the line (class ⭘)

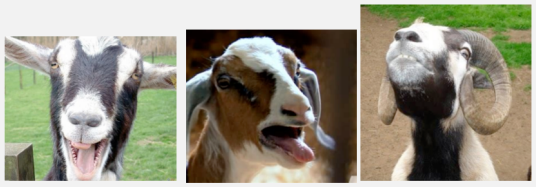
Answer: $x_2 - x_1 - 1 = 3 - 4 - 1 = -2 < 0$
$\Rightarrow$ classify point as ⭘

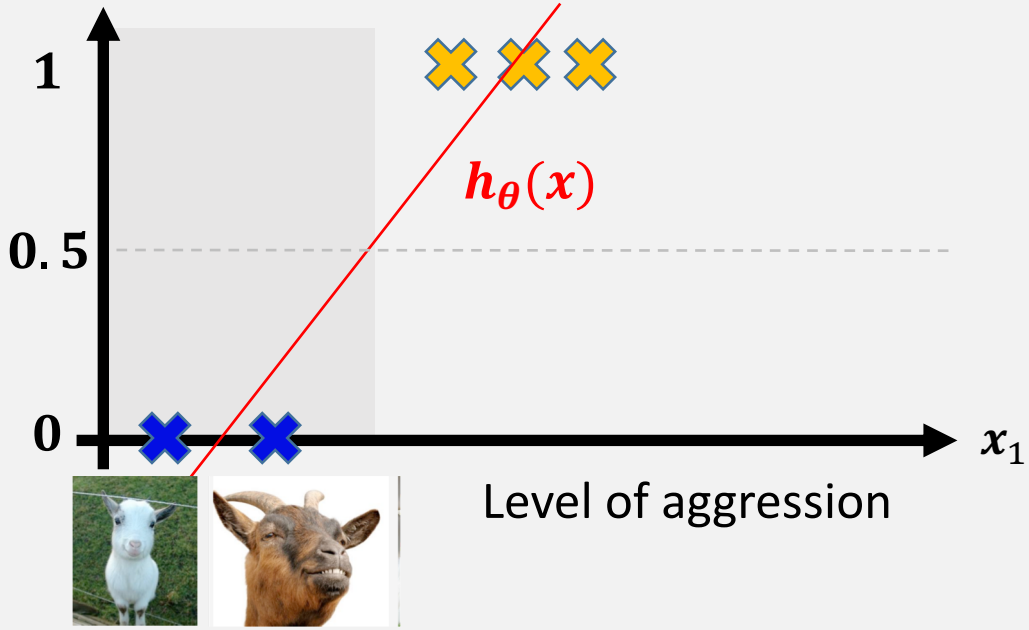
But how do we find this line?

Angry Goat Classification $y \in \{0,1\}$

Apply linear regression with hypothesis $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1$ (with $x_0 = 1$)

- If $h_\theta(x) \geq 0.5$, predict $y = 1$ for x
- Else predict $y = 0$

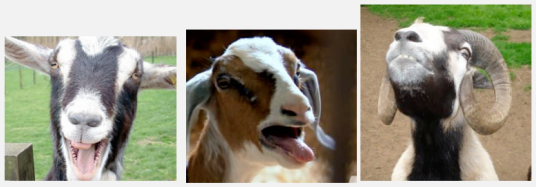


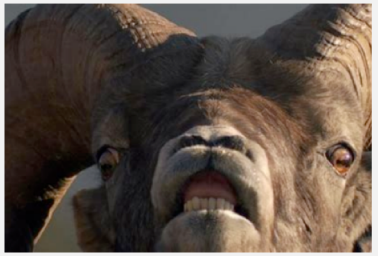


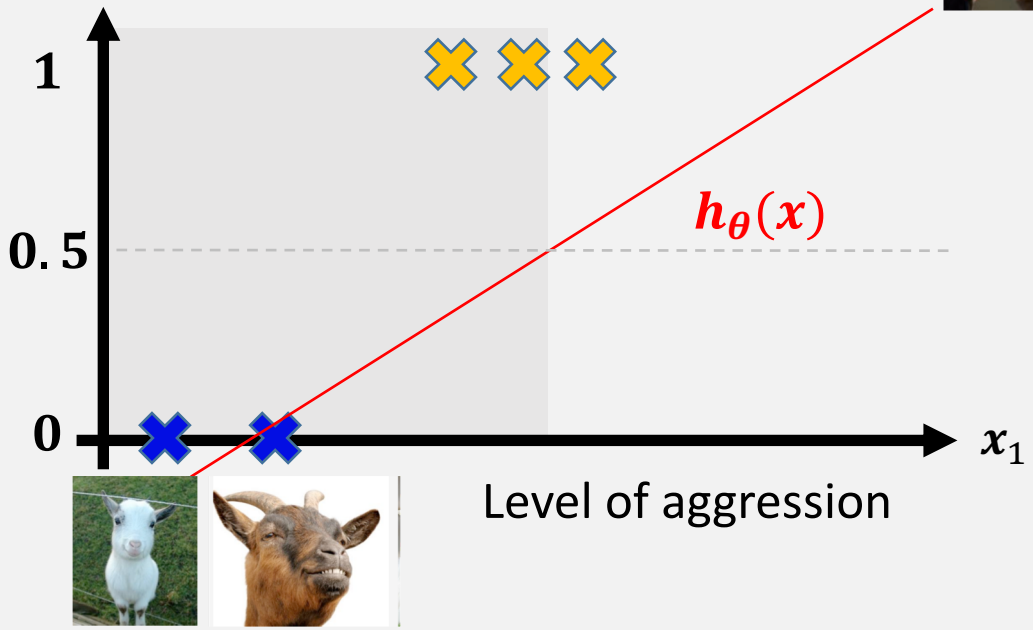
Angry Goat Classification $y \in \{0,1\}$

Apply linear regression with hypothesis $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1$ (with $x_0 = 1$)

- If $h_\theta(x) \geq 0.5$, predict $y = 1$ for x
- Else predict $y = 0$







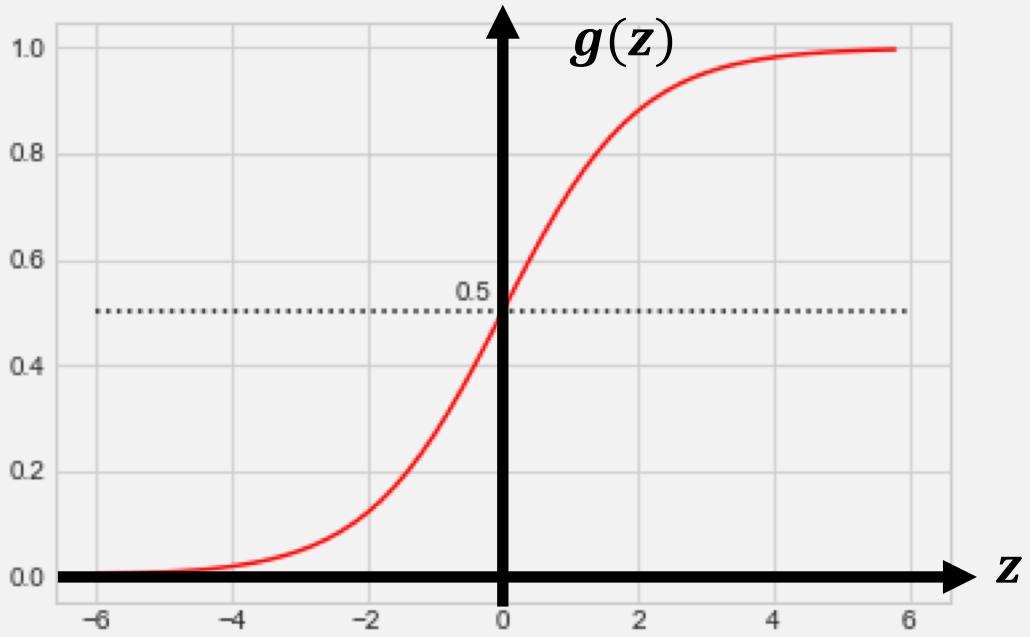
- Not well behaved
- Hypothesis function is not constrained to lie between 0 and 1

The logistic (or sigmoid) function

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

$$\boldsymbol{0 \leq g(z) \leq 1}$$

If $z = 0$: $g(0) = 0.5$
If $z \gg 0$: $g(z) \to 1$
If $z \ll 0$: $g(z) \to 0$



Logistic Regression (actually classification!)

Use the linear hypothesis function (from linear regression) as the argument to the logistic function

- Linear regression $h_\theta(x) = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \theta^T x$
- Logistic function $g(z) = \frac{1}{1+e^{-z}}$

Logistic regression

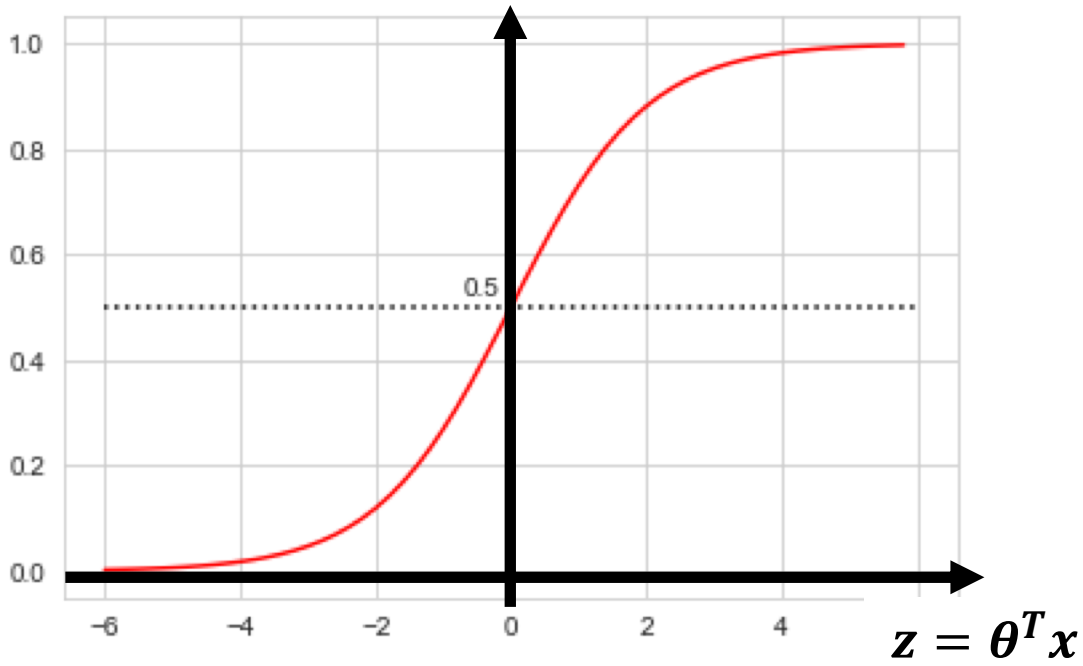
$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Logistic regression

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

This is a probability

- Range of [0,1]
- $h_\theta(x) = P(y = 1 \mid x; \theta)$
- i.e. the probability that $y = 1$ given x and parameterized by θ

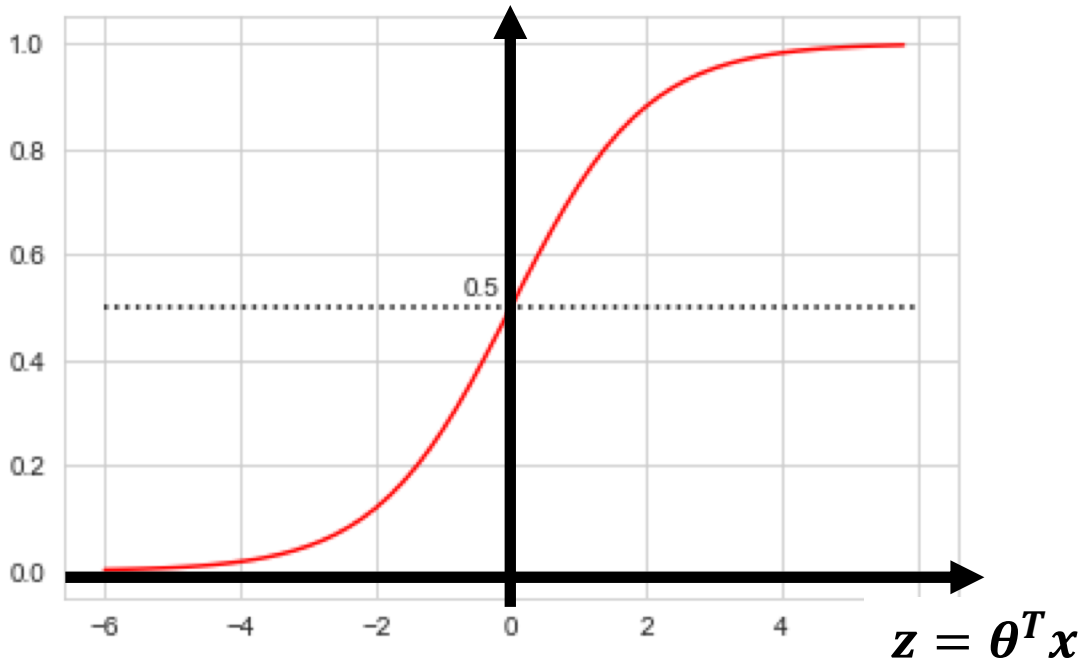


Logistic regression

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

This is a probability

- Range of [0,1]
- $h_\theta(x) = P(y = 1 \mid x; \theta)$
- i.e. the probability that $y = 1$ given x and parameterized by θ

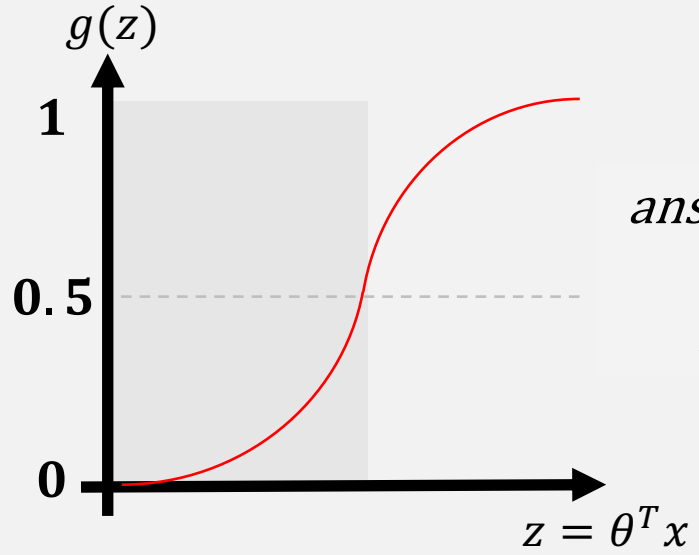


Decision?

- If $h_\theta(x) \geq 0.5$: $predict\ y = 1$
- If $h_\theta(x) < 0.5$: $predict\ y = 0$

Logistic regression: decision boundary

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



answer no (y=0)
$\theta^T x < 0$
$h_\theta(x) < 0.5$

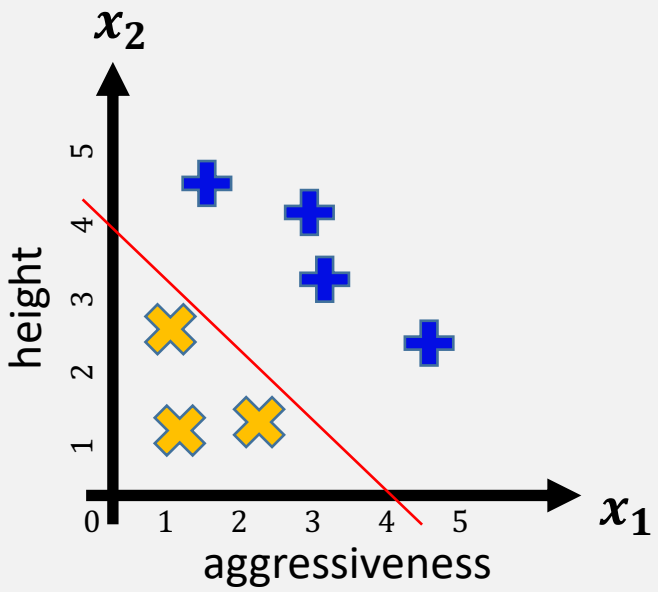
answer yes (y=1)
$\theta^T x \geq 0$
$h_\theta(x) \geq 0.5$

Decision boundary

- Separates $y = 0$ from $y = 1$, as created by our hypothesis function

Decision boundary: example 1

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



Given hypothesis function

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

with $\theta_0 = -4,\ \theta_1 = 1, \theta_2 = 1,$

what does the decision boundary look like?

→ a line, i.e. $\theta_0 + \theta_1 x_1 + \theta_2 x_2$

→ everything above is $y = 1$, below $y = 0$

Decision boundary:

→ $x_2 + x_1 - 4 \geq 0$: $h_\theta(x) \geq 0.5$

predict $y = 1$ ( )

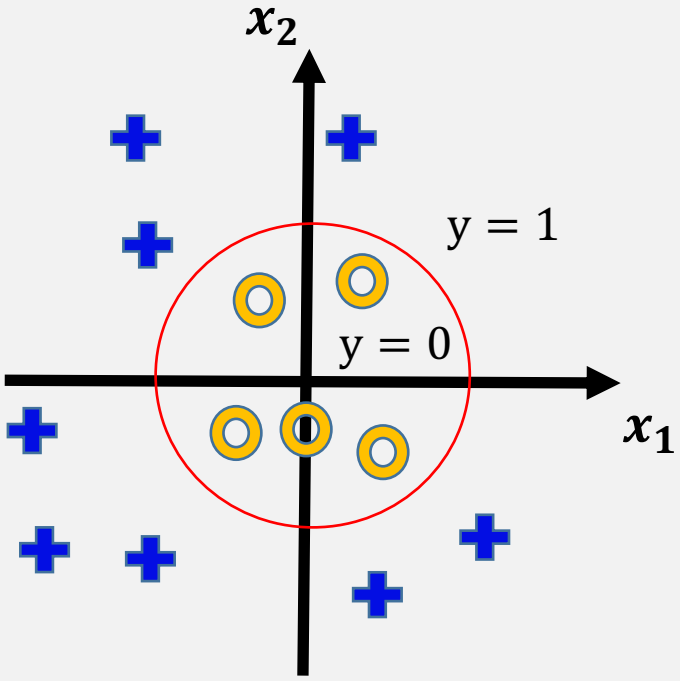
→ $x_2 + x_1 - 4 < 0$: $h_\theta(x) < 0.5$

predict $y = 0$ ( )

Because the input to the logistic function is a line, then the decision boundary is also a line

Decision boundary: example 2

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



Given hypothesis function

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

with $\theta_0 = -1,\ \theta_1 = 1, \theta_2 = 1,$

($x_2^2 + x_1^2 - 1$ is an equation of a circle)

Decision boundary:

→ $x_2^2 + x_1^2 - 1 \geq 0$: $h_\theta(x) \geq 0.5$

predict y = 1 (✚)

→ $x_2^2 + x_1^2 - 1 < 0$: $h_\theta(x) < 0.5$

predict y = 0 (○)

Because the input to the logistic function is a circle, then the decision boundary is also a circle

Logistic regression and probability

The hypothesis can be treated as a probability

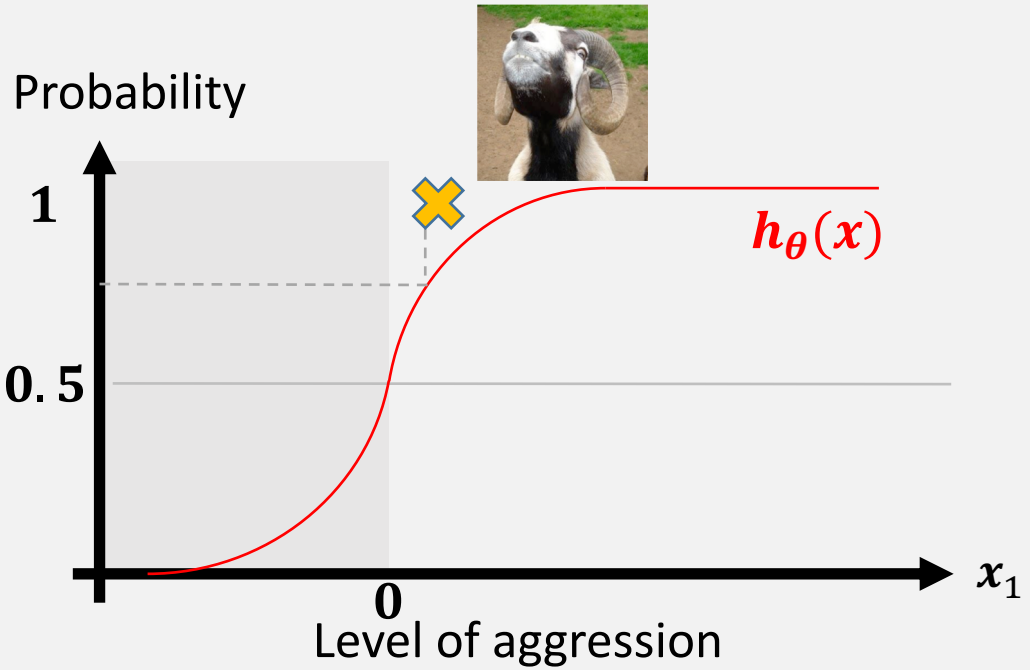
- if $h_\theta(x) = 0.75$, there is a 75% probability of class 1 (angry goat)
- What is the corresponding probability of it **not** being angry?

$P(not\ angry|\text{✖}) = 1 - 0.75 = 0.25$

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

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

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



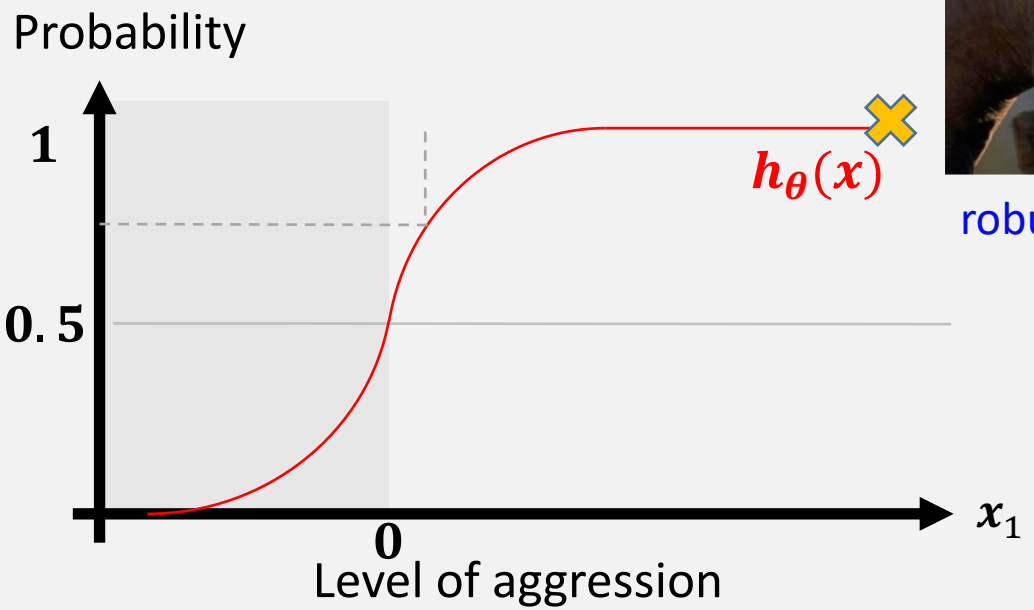
Logistic regression and probability

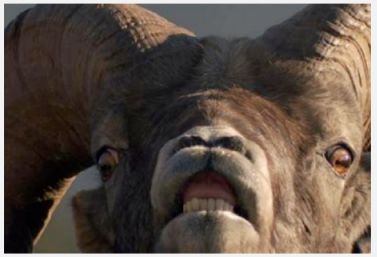
The hypothesis can be treated as a probability

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

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

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

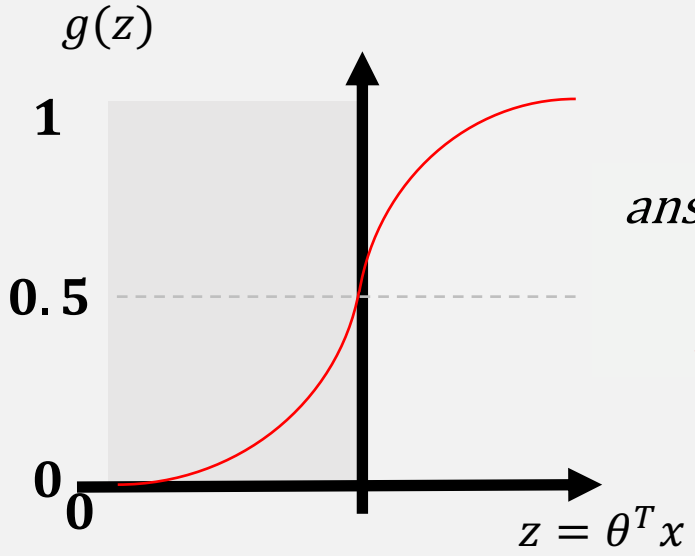




robust to outliers

Logistic regression: decision boundary

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



answer no (y=0)
$\theta^T x < 0$
$h_\theta(x) < 0.5$

answer yes (y=1)
$\theta^T x \geq 0$
$h_\theta(x) \geq 0.5$

How do we find the optimal parameters θ for logistic regression?

Lecture 6: Logistic Regression

Logistic regression

- Regression and classification
- The sigmoid (logistic) function
- Optimisation for logistic regression

Further issues

- Multi-class classification
- Overfitting and Underfitting
- Regularisation

Optimisation for Logistic Regression

Using training dataset: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots, (x^{(m)}, y^{(m)})\}$

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

(where $\theta^T x = \sum_{j=0}^{n} \theta_j x_j^{(i)}$)

Question: How do we find the best parameters θ for Logistic Regression classification?

Linear Regression: reminder

1. Hypothesis function (a multi-dimensional line)

$$h_\theta\left(x^{(i)}\right) = \sum_{j=0}^{n} \theta_j x_j^{(i)} = \theta^T x \qquad \text{(where } x_0^{(i)} = 1\text{)}$$

2. Loss (least squares)

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

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

3. Gradient descent update rule

while not converged:

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

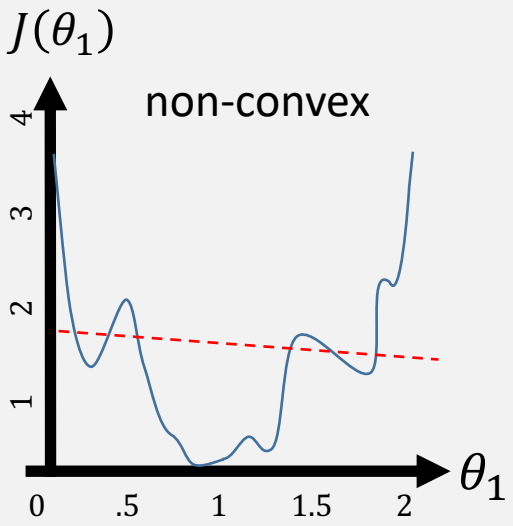
for $j = 0,1, \ldots, n$

$(x_0 = 1)$

Optimisation for Logistic Regression

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

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



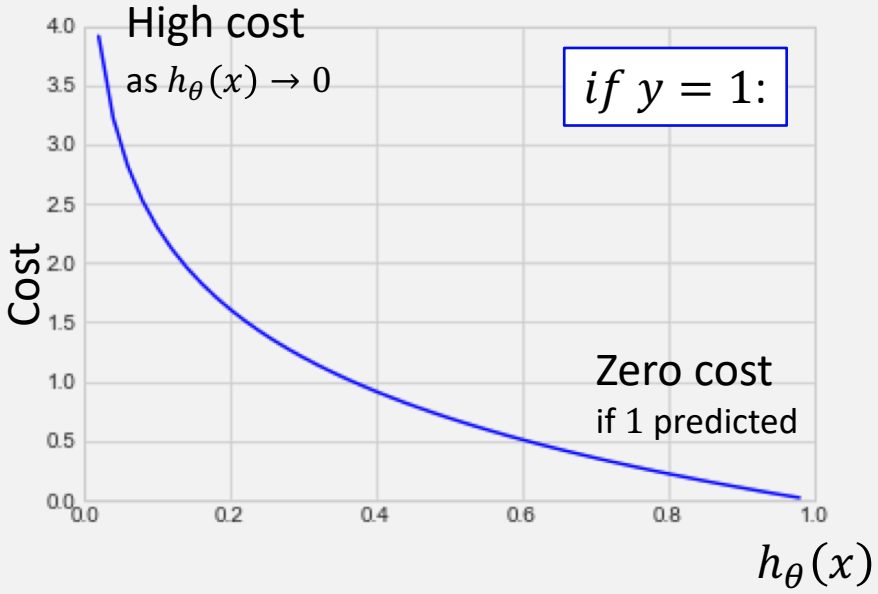
This loss function is non-convex with respect to θ

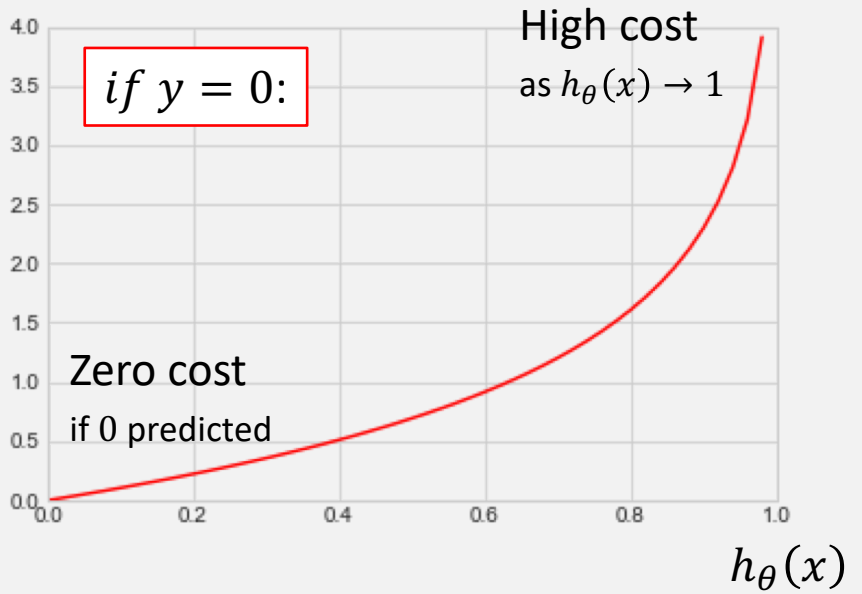
- Many local optima
- Cannot guarantee finding global optimum
- We need an alternative cost function

An alternative Cost Function: Cross Entropy

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

This cost function is **convex** with respect to θ
→ We can always find a global minimum using gradient descent





Intuition: heavily penalise algorithm if correct answer is 1, but it outputs 0.

Intuition: heavily penalise algorithm if correct answer is 0, but it outputs 1.

An alternative Cost Function: Cross Entropy

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

This cost function is **convex** with respect to θ
→ We can always find a global minimum using gradient descent

How might we write this cost function as one equation to use in our gradient descent loss function?

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

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

becomes 0 if $y = 0$

becomes 0 if $y = 1$

Logistic Regression: gradient descent

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

$$Cost(h_\theta(x), y) = -ylog\big(h_\theta(x)\big) - (1-y)\log(1-h_\theta(x))$$

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

Gradient descent update

while not converged:

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

for $j = 0,1, \dots, n$

Logistic Regression: gradient descent

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

$$Cost(h_\theta(x), y) = -ylog\big(h_\theta(x)\big) - (1-y)\log(1-h_\theta(x))$$

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

Using the chain rule…

$$\frac{\partial}{\partial \theta_j} Cost = \frac{\partial Cost}{\partial h_\theta}\frac{\partial h_\theta}{\partial(\theta^T x)}\frac{\partial(\theta^T x)}{\partial \theta_j}$$

$$= \left(\left(-\frac{y}{h_\theta} + \frac{1-y}{1-h_\theta}\right)(1-h_\theta)h_\theta\right)x_j$$

$$= (-y + yh_\theta + h_\theta - yh_\theta)x_j$$

$$= (h_\theta - y)x_j$$

$\frac{\partial \log(x)}{\partial x} = \frac{1}{x}$

Seem familiar?

(you can skip this bit!)

Logistic Regression: gradient descent

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

$$Cost(h_\theta(x), y) = -ylog\big(h_\theta(x)\big) - (1-y)\log(1-h_\theta(x))$$

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

Gradient descent update

while not converged:

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

$$= \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m} \big(h_\theta\big(x^{(i)}\big) - y^{(i)}\big)\, x_j^{(i)} \qquad \text{with } x_0^{(i)} = 1$$

This is identical to the update rule for linear regression
→ The only change is the hypothesis function

Optimisation for Logistic Regression

Using training dataset: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \dots, (x^{(m)}, y^{(m)})\}$

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

(where $\theta^T x = \sum_{j=0}^{n} \theta_j x_j^{(i)}$)

with $x_0^{(i)} = 1$

Question: How do we find the best parameters θ for Logistic Regression classification?

Answer: Gradient descent

while not converged:

$$\theta_j^{new} = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x_j^{(i)} \quad \text{for } j = 0,1, \dots, n$$

with $x_0^{(i)} = 1$

Note: There is no closed-form (normal equation) solution for logistic regression

Quiz: Logistic Regression

You are put in charge of the cake display at a supermarket. There is a database containing the following historical cake information:

- y: is the cake edible or not? *{1: edible, 0: inedible}*
- x_1: time since baked (hours)
- x_2: temperature of cake display (degrees Celsius)

Design a logistic regression classifier that predicts whether a cake is edible or not.

1. Write a hypothesis, h_θ (using $g(z)$)

$$h_\theta^{LR} = h(\boldsymbol{x} = \{x_0, x_1, x_2\}; \theta) = g(\theta^T \boldsymbol{x}) = g\left(\sum_{i=0}^{2} \theta_i \, x_i\right) = g(\, \theta_0 + \theta_1 x_1 + \theta_2 x_2\,)$$

with $x_0 = 1$

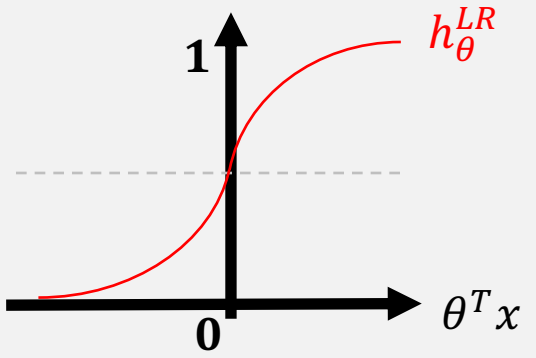
2. What is the decision function?

If $h_\theta^{LR}(x_1, x_2) \geq 0.5$: edible, otherwise inedible

Logistic Regression (single variable)

1. Hypothesis (for data sample $x^{(i)}$)

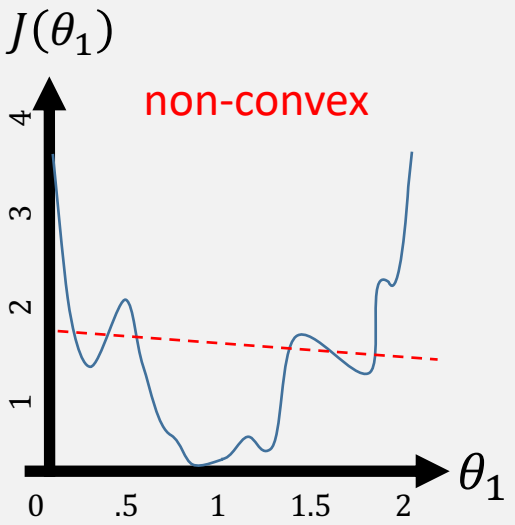
$$h_\theta^{LR}(x^{(i)}) = g(\theta_0 + \theta_1 x^{(i)})$$
$$= g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$



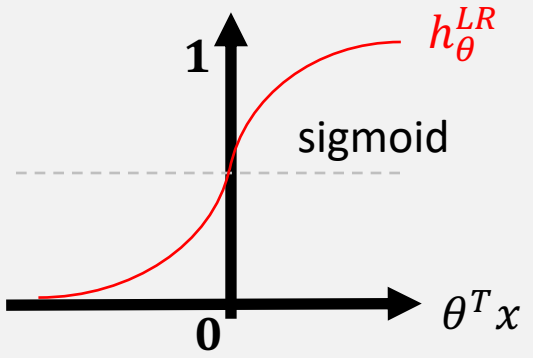
2. try (L2) Loss over m data samples

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

=> Not linear in θ => non-convex => no global minimum



Logistic Regression (single variable)



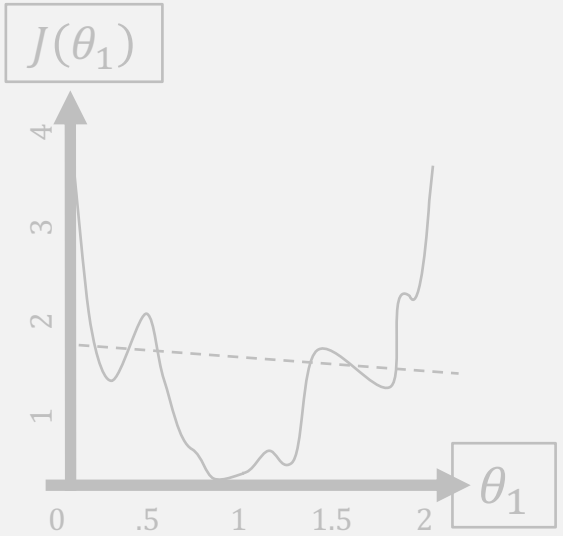
1. Hypothesis (for data sample $x^{(i)}$)

$$h_\theta^{LR}(x^{(i)}) = g(\theta_0 + \theta_1 x^{(i)})$$
$$= g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

2. try (L2) Loss over m data samples

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

=> Not linear in θ => non-convex => no global minimum



2. Alternative loss (use logarithms)

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m} Cost(h_\theta^{LR}(x^{(i)}), y^{(i)})$$

$$with\ Cost(h_\theta^{LR}, y) = -y \cdot log(h_\theta^{LR}) - (1-y) \cdot \log(1 - h_\theta^{LR})$$

More resources on logistic regression

- Example from Wikipedia page on logistic regression
  - **“Probability of passing an exam versus hours of study”**

- Have a look at Andrew Ng’s lectures on the topic
  (free Coursera course & on youtube)

Lecture 6: Logistic Regression

Logistic regression

- Regression and classification
- The sigmoid (logistic) function
- Optimisation for logistic regression

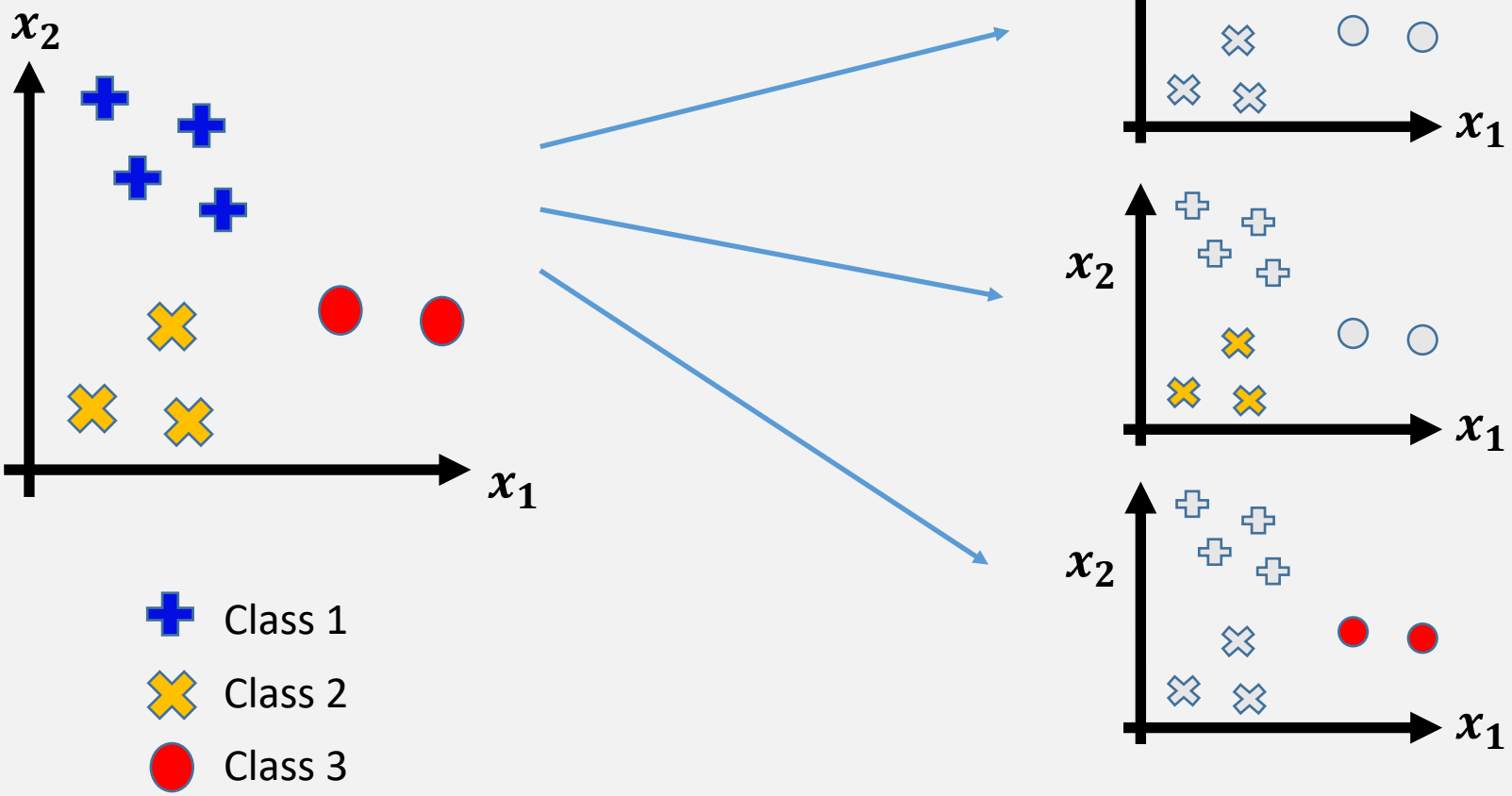
Further issues

- Multi-class classification
- Overfitting and Underfitting
- Regularisation

Multi-class classification

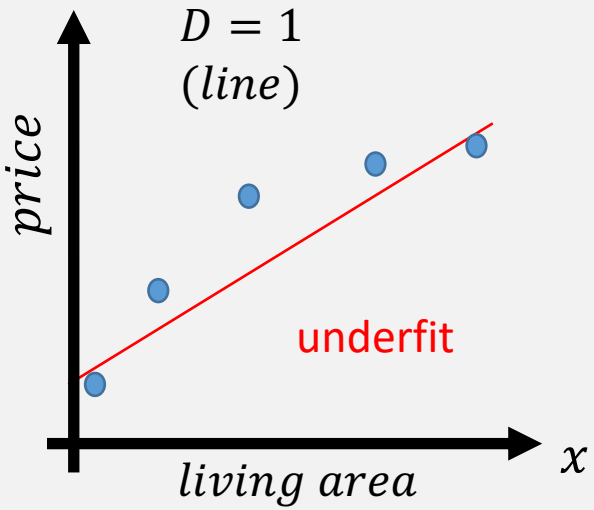
Given a binary classifier ($y = \{0,1\}$), how would we do multi-class classification?

- One-vs-all (one-vs-rest) classification

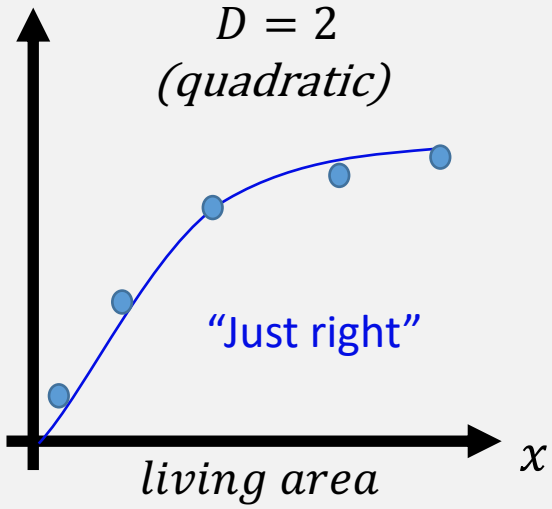


Overfitting in polynomial regression

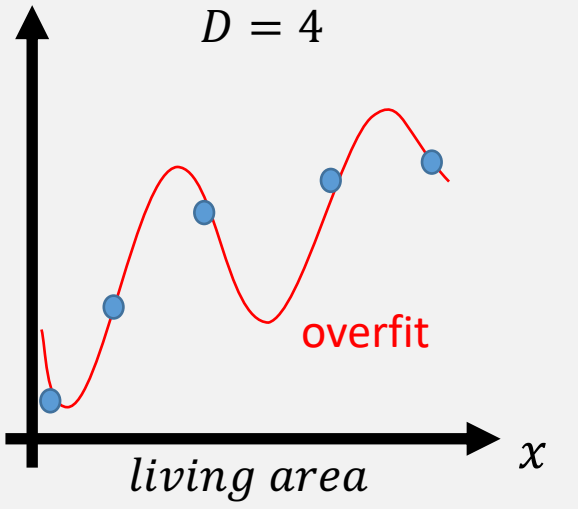
For hypothesis functions with varying polynomial degrees, D



$\theta_0 + \theta_1 x$



$\theta_0 + \theta_1 x + \theta_2 x^2$

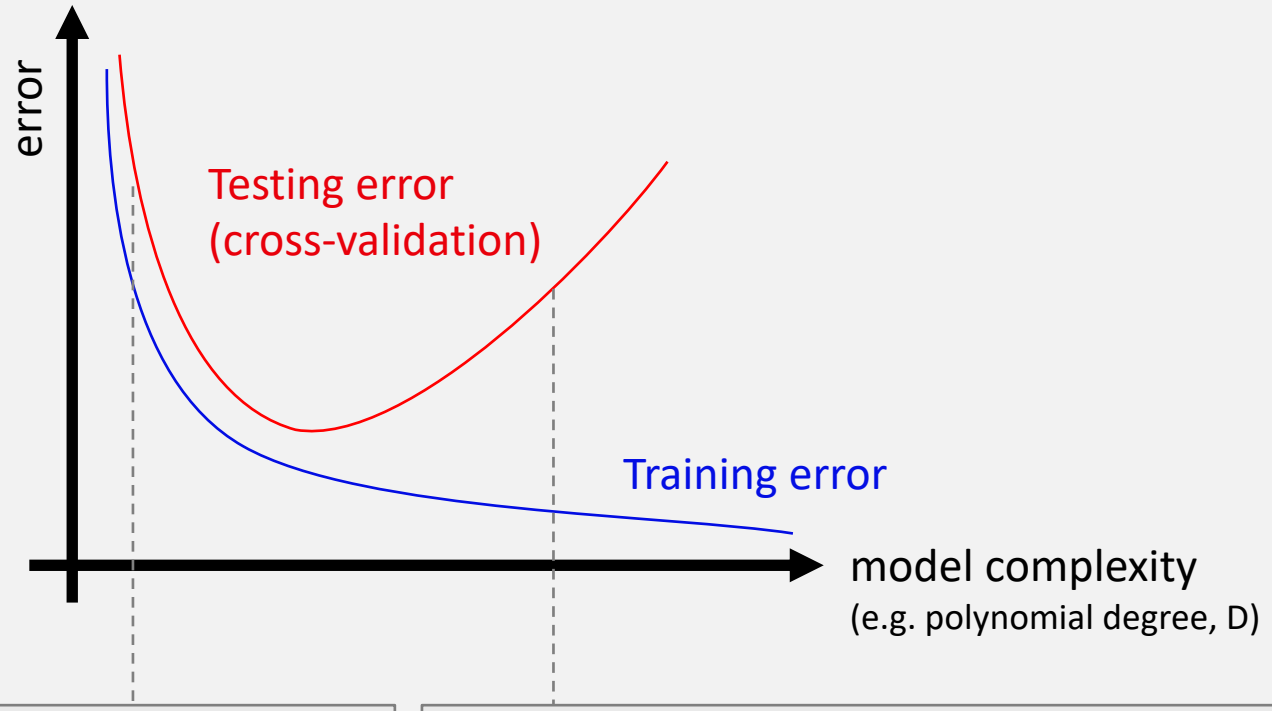


$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High bias: the model parameters are *biased* (are overly simple and inflexible)

High variance: the model parameters *vary* a lot (overly complex and "wavy")

Overfitting: cross validation error



Underfitting (high bias)

- ▸ hypothesis maps poorly to the data
  - model too simple, or
  - too few features

Overfitting (high variance)

- ▸ hypothesis fits data well, but does not generalize well to unseen data
  - model too complex, or
  - too many features

Generalisation: How well a hypothesis predicts new data

Solutions to overfitting

1. Reduce the number of features
   - ▸ Manually select features to keep
   - ▸ Use a model selection algorithm (e.g. cross validation)

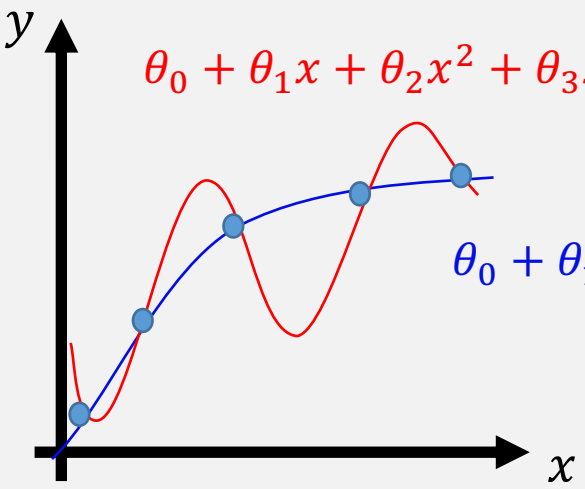
2. Regularisation
   - ▸ Keep all the features, but reduce the number of parameters
   - ▸ Works well when we have lots of features contributing to prediction

Regularisation

Reduces overfitting by keeping the parameter values low

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

- Models with D=2 and D=4 both have the zero loss
- But D4 is obviously overfitting



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$ $\leftarrow \theta = [1.8 \quad 4.2 \quad -2 \quad 0.5 \quad -0.1]^T$

$J(\theta) = 0$

$\sum \theta^2 \cong 25$

$\theta_0 + \theta_1 x + \theta_2 x^2$ $\leftarrow \theta = [-0.1 \quad 1.1 \quad -0.1]^T$

$J(\theta) = 0$

$\sum \theta^2 \cong 1.2$

Solution: if we add a penalty to the loss function that is related to the size of θ, we can penalise complexity in the model

- e.g. a simple penalty term might be θ^2 (*used in L2 regularisation*)

Regularisation

Add a penalty term to the loss function to penalise complexity:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} Cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

Note, we do not penalise θ_0 because it corresponds to $x_0 = 1$

Any algorithm that tries to find the parameters θ that minimise this new loss function, i.e. $min_\theta J(\theta)$, will favour lower values for θ

Regularisation parameter λ

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} Cost\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right)\right] + \frac{\lambda}{2}\sum_{j=1}^{n} \theta_j^2$$

The regularisation parameter determines how much we inflate the cost of parameters θ in the loss function

If λ too big?

- algorithm underfits (θ very small)

If λ too small?

- algorithm may overfit (θ can be very large)

Optimising regularised functions

Batch Gradient Descent (for both linear & logistic regression)

Note, we do not regularise θ_0, so this is the same as before

while not converged:

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

$$\theta_j^{new} = \theta_j - \frac{\alpha}{m} \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) x_j^{(i)} \right] + \lambda \theta_j$$

for $j = 1, \ldots, n$

Regularised Normal Equation (a.k.a. Ridge Regression)

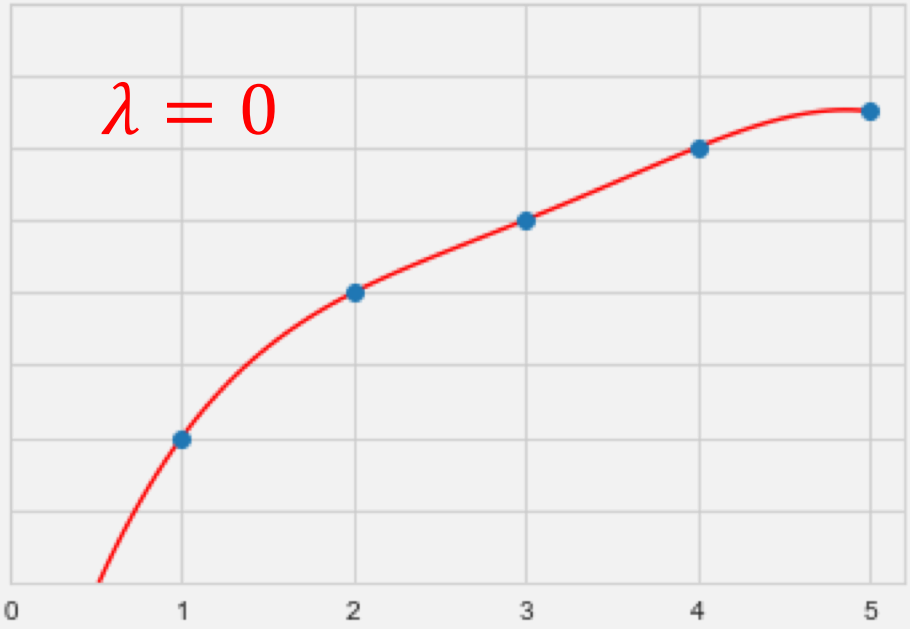
$$\theta = (X^T X + \lambda E)^{-1} X^T y$$

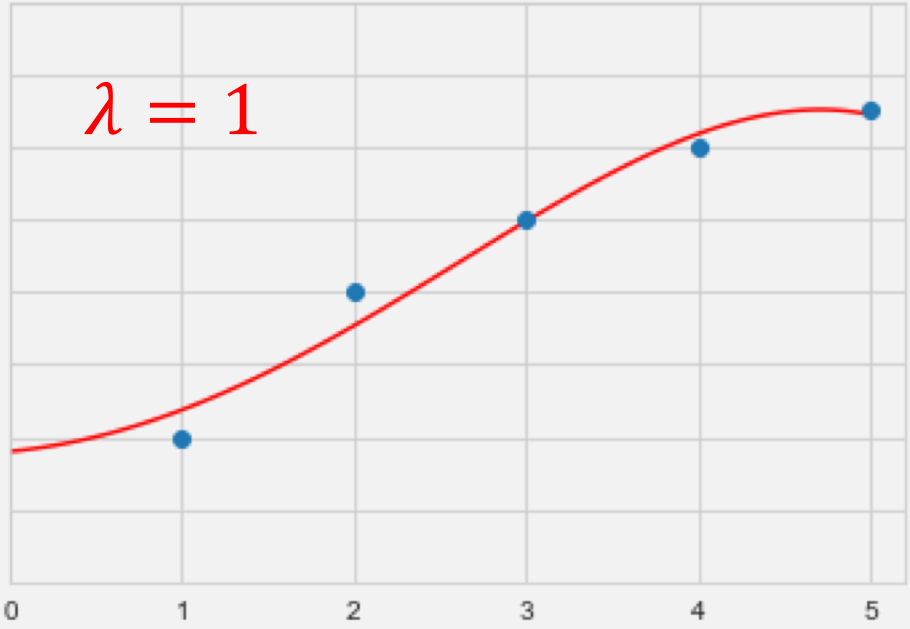
Normal Equation
$\theta = (X^T X)^{-1} X^T y$

Where $E = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}$

→ almost the $(n+1) \times (n+1)$ identity matrix but with the first element zero (so that we don't regularise the x_0 term)

Regularisation example





$\theta = [-1.75 \quad 4.23 \quad -1.84 \quad 0.4 \quad -0.03\ ]$

$\theta = [\ 0.9 \quad 0.1 \quad 0.2 \quad -0.001 \quad -0.001 \quad -0.01]$

$$\sum_{j=1} \theta_j^2 \approx 21$$

Loss, $J(\theta) = 0$

$$\sum_{j=1} \theta_j^2 \approx 0.05$$

Loss, $J(\theta) = 0.02$

Regularisation simplifies the model by reducing the effect of higher order terms

Summary

- Logistic regression
  - When you want a binary classifier (with estimation probability)
  - Like linear regression, but use logistic (sigmoid) in hypothesis
  - Gradient Descent optimisation is identical to linear regression

- Overfitting and Underfitting
  - High bias: inflexible model, underfits data
  - High variance: model values vary a lot, overfits data
  - We want a generalisable model, balancing bias with variance

- Regularisation
  - An alternative to reducing feature count
  - Penalise model complexity by adding penalty term to the loss function