Linear Regression

CISC 7026 - Introduction to Deep Learning

Steven Morad

University of Macau

Announcements

Announcements

Homework 0 was ok?

Homework 1 released, due in 2 weeks (see Moodle)
- Discuss more at the end of class

I will be away 09.12 and 09.19
- Yutao will lecture 09.12
- My students will proctor exam 1 on 09.19

Announcements

Currently writing exam 1, probably 6 questions:

- 1 question function notation
- 1 question set notation
- 2 questions linear regression (make sure you can invert 2x2 matrices)
- 1 question neural networks (neurons)
- 1 question gradient descent (know how to take derivatives, no need to memorize formulas)

Bring a pen/pencil/eraser to exam, you need nothing else

You will have 3 hours to finish the exam

- Probably takes most students 1-1.5 hours
- No rush, take as long as you need

Review

Review

We often know **what** we want, but we do not know **how**

We have many pictures of either dogs or muffins $x \in X$

We want to know if the picture is [dog | muffin] $y \in Y$

We learn a function or mapping from X to Y

$$f : X \times \mapsto Y$$

Review

Usually, functions are defined once and static: $f(x) = x^2$

But in machine learning, we must **learn** the function

To avoid confusion, we introduce the **function parameters**

$$\theta \in \Theta \qquad\qquad f : X \times \Theta \mapsto Y$$

Math Notation

Math Notation - Sets

Before we go any futher, we need to agree on math notation

If you ever get confused, come back to these slides

Vectors bold small characters $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

Matrices bold big characters $\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}$

Math Notation - Sets

We will represent **tensors** as nested vectors or matrices

Tensor

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix}$$

Each $\boldsymbol{x}_i$ is a vector

Math Notation - Sets

Same for matrices

Tensor of matrices

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_{1,1} & \boldsymbol{x}_{1,2} & \dots & \boldsymbol{x}_{1,n} \\ \boldsymbol{x}_{2,1} & \boldsymbol{x}_{2,2} & \dots & \boldsymbol{x}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{m,1} & \boldsymbol{x}_{m,2} & \dots & \boldsymbol{x}_{m,n} \end{bmatrix}$$

I use square brackets for data index

$x_{[i],j,k}$ indexes a 3D tensor, where the first dimension is the dataset

- Dataset of matrices (2D)

Math Notation - Sets

Question: What is the difference between the following?

$$\boldsymbol{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}$$

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_{1,1} & \boldsymbol{x}_{1,2} & \cdots & \boldsymbol{x}_{1,n} \\ \boldsymbol{x}_{2,1} & \boldsymbol{x}_{2,2} & \cdots & \boldsymbol{x}_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_{m,1} & \boldsymbol{x}_{m,2} & \cdots & \boldsymbol{x}_{m,n} \end{bmatrix}$$

Math Notation - Sets

Exception: I will always write the parameters θ lowercase

- We introduce a scalar parameter θ
- Then introduce a parameter vector $\boldsymbol{\theta}$
- But later it becomes a matrix $\boldsymbol{\theta}$
- And then a tensor (vector of matrices) $\boldsymbol{\theta}$

Math Notation - Sets

Capital letters will often refer to **sets**

$$X = \{1, 2, 3, 4\}$$

We will represent important sets with blackboard font

$\mathbb{R}$	Set of all real numbers $\{-1, 2.03, \pi, ...\}$
$\mathbb{Z}$	Set of all integers $\{-2, -1, 0, 1, 2, ...\}$
$\mathbb{Z}_+$	Set of all **positive** integers $\{1, 2, ...\}$

Math Notation - Sets

$[0, 1]$	Closed interval $0.0, 0.01, 0.00...1, 0.99, 1.0$
$(0, 1)$	Open interval $0.01, 0.00...1, 0.99$
$\{0, 1\}$	Set of two numbers (boolean)
$[0, 1]^k$	A vector of k numbers between 0 and 1
$\{0, 1\}^{k \times k}$	A matrix of boolean values of shape k by k

Math Notation - Sets

We will use various set operations

$A \subseteq B$	A is a subset of B
$A \subset B$	A is a strict subset of B
$a \in A$	a is an element of A
$b \notin A$	b is not an element of A
$A \cup B$	The union of sets A and B
$A \cap B$	The intersection of sets A and B

Math Notation - Sets

We will often use **set builder** notation

$$\{ \underbrace{x + 1}_{\text{Function}} \mid \underbrace{x \in \{1, 2, 3, 4\}}_{\text{Domain}} \}$$

You can think of this as a for loop

```
output = {} # Set
for x in {1, 2, 3, 4}:
  output.insert(x + 1)
```

```
output = {x + 1 for x in {1, 2, 3, 4}}
```

Math Notation - Functions

We define **functions** or **maps** between sets

$$\underset{\text{Name}}{f} : \underset{\text{Input}}{\mathbb{R}} \mapsto \underset{\text{Output}}{\mathbb{Z}}$$

This function f maps a real number to an integer

Question: What functions could f be?

$$\text{round} : \mathbb{R} \mapsto \mathbb{Z}$$

Math Notation - Functions

Functions can have multiple inputs

$$f : X \times \Theta \mapsto Y$$

The function f maps elements from sets X and Θ to set Y

I will define variables when possible

$$X = \mathbb{R}^n \qquad \Theta = \mathbb{R}^{m \times n} \qquad Y = [0, 1]^{n \times m}$$

Math Notation - Functions

Finally, functions can have a function as input or output

Question: Any examples?

$$\frac{\mathrm{d}}{\mathrm{d}x} : \underbrace{(f : \mathbb{R} \mapsto \mathbb{R})}_{\text{Input function}} \mapsto \underbrace{(f' : \mathbb{R} \mapsto \mathbb{R})}_{\text{Output function}}$$

$$\frac{\mathrm{d}}{\mathrm{d}x}[x^2] = 2x$$

Notation Exercises

Notation Exercises

$\mathbb{R}^n$	Set of all vectors containing n real numbers
$\{3, 4, ..., 31\}$	Set of all integers between 3 and 31
$[0, 1]^n$	Set of all vectors of length n with values between 0 and 1
$\{0, 1\}^n$	Set of all boolean vectors of length n

Notation Exercises

$$\left\{x^{\frac{1}{2}} \mid x \in \mathbb{R}_{+}\right\}$$

Question: What is this?

Answer: The results of evaluating $f(x) = \sqrt{x}$ for all positive real numbers

$$\{2x \mid x \in \mathbb{Z}_{+}\}$$

Question: What is this?

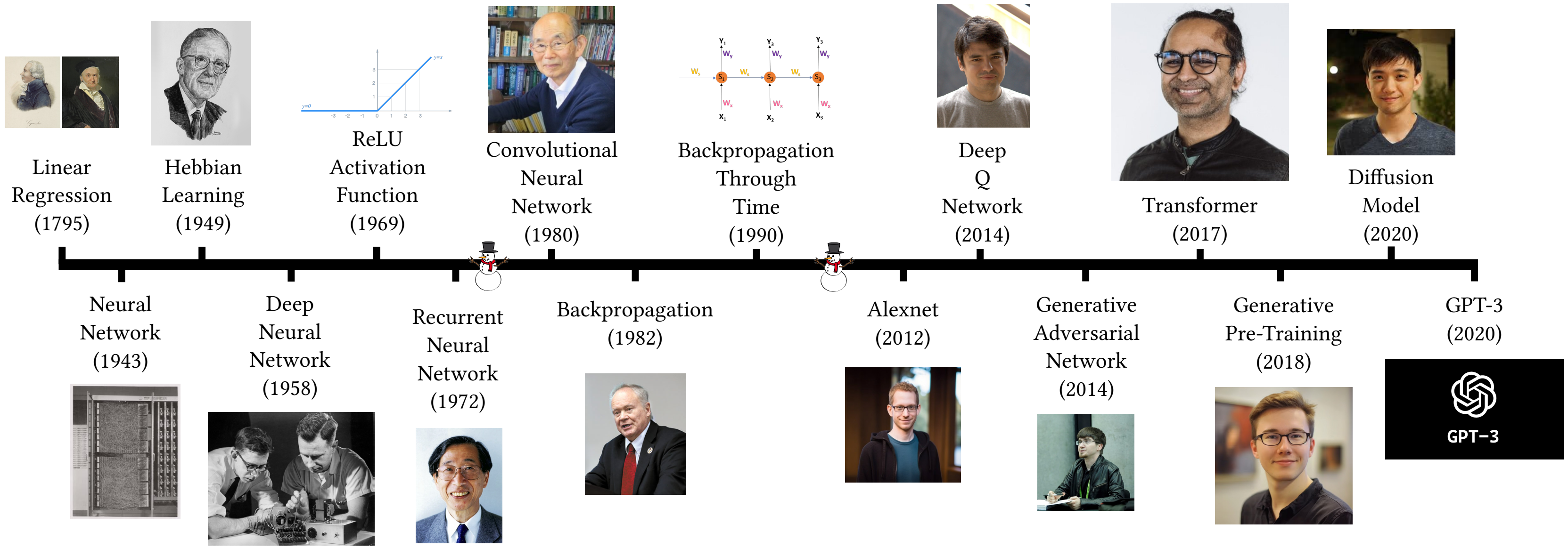
Answer: Set of all positive even integers

Linear Regression

Linear Regression

Today, we will learn about linear regression

Probably the oldest method for machine learning (Gauss and Legendre)



Neural networks share many similarities with linear regression

Linear Regression

Many problems in ML can be reduced to **regression** or **classification**

Regression asks how many

- Given my parents height, How tall will I be?
- Given the rain today, how much rain will there be tomorrow?
- Given a camera image, how far away is this object?

Classification asks which one

- Is this image of a dog or muffin?
- Given the rain today, will it rain tomorrow? Yes or no?
- Given a camera image, what color is this object? Yellow, blue, red?

Let us start with regression

Linear Regression

Today, we will come up with a regression problem and then solve it!

Remember the four steps of machine learning

1. Define an example problem and dataset
2. Define our linear model f
3. Define a loss function $\mathcal{L}$
4. Find parameters using $\mathcal{L}$ (optimization)

We will combine these to solve the example problem

Linear Regression - Example Problem

The World Health Organization (WHO) collected data on life expectancy

Available for free at https://www.who.int/data/gho/data/themes/mortality-and-global-health-estimates/ghe-life-expectancy-and-healthy-life-expectancy

Linear Regression - Example Problem

The data comes from roughly 3,000 people from 193 countries

For each person, they recorded:

- Home country
- Alcohol consumption
- Education
- Gross domestic product (GDP) of the country
- Immunizations for Measles and Hepatitis B
- How long this person lived

We can use this data to make future predictions

Linear Regression - Example Problem

Since everyone here is very educated, we will focus on how education affects life expectancy

There are studies showing a causal effect of education on health

- *The causal effects of education on health outcomes in the UK Biobank.* Davies et al. *Nature Human Behaviour.*
- By staying in school, you are likely to live longer

Linear Regression - Example Problem

Task: Predict life expectancy

$X = \mathbb{R}_+$: Years in school

$Y = \mathbb{R}_+$: Age of death

Each $x \in X$ and $y \in Y$ represent a single person

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

Goal: Given someone's education, predict how long they will live

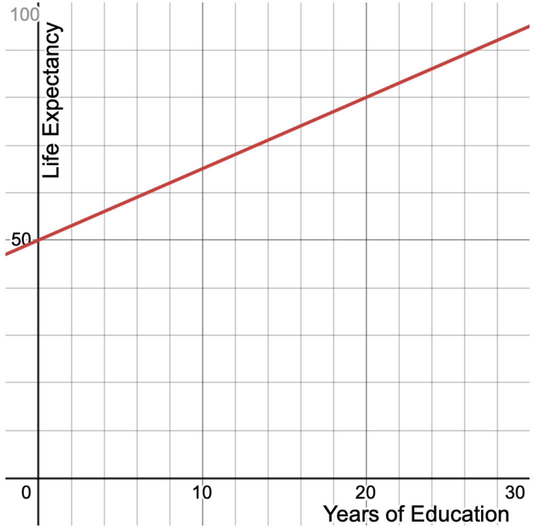
Linear Regression - Model

Soon, f will be a deep neural network

The core of all neural networks are **linear functions**

For now, we let f be a linear function

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}\right) = \theta_0 + \theta_1 x$$



Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ that makes $f(x, \boldsymbol{\theta}) = y$

Linear Regression - Loss Function

Now, we need to find the parameters $\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix}$ that make $f(x, \boldsymbol{\theta}) = y$

Question: How do we choose $\boldsymbol{\theta}$?

Answer: $\boldsymbol{\theta}$ that makes $f(x, \boldsymbol{\theta}) = y; \quad x \in X, y \in Y$

1. Find a loss function
2. Choose $\boldsymbol{\theta}$ with the smallest loss

Linear Regression - Loss Function

The loss function computes the loss for the given parameters and data

$$\mathcal{L} : X^n \times Y^n \times \Theta \mapsto \mathbb{R}$$

The loss function should tell us how close $f(x, \boldsymbol{\theta})$ is to y

By **minimizing** the loss function, we make $f(x, \boldsymbol{\theta}) = y$

There are many possible loss functions, but for regression we often use the **square error**

$$\text{error}(y, \hat{y}) = (y - \hat{y})^2$$

Linear Regression - Loss Function

We can write the loss function for a single datapoint $x_{[i]}, y_{[i]}$ as

$$\mathcal{L}\left(x_{[i]}, y_{[i]}, \boldsymbol{\theta}\right) = \text{error}\left(f\left(x_{[i]}, \boldsymbol{\theta}\right), y_{[i]}\right) = \left(f\left(x_{[i]}, \boldsymbol{\theta}\right) - y_{[i]}\right)^2$$

Question: Will this $\mathcal{L}$ give us a good prediction for all possible x?

Answer: No! We only consider a single datapoint $x_{[i]}, y_{[i]}$. We want to learn $\boldsymbol{\theta}$ for the entire dataset, for all $x \in X, y \in Y$

Linear Regression - Loss Function

For a single $x_{[i]}, y_{[i]}$:

$$\mathcal{L}\left(x_{[i]}, y_{[i]}, \boldsymbol{\theta}\right) = \text{error}\left(f\left(x_{[i]}, \boldsymbol{\theta}\right), y_{[i]}\right) = \left(f\left(x_{[i]}, \boldsymbol{\theta}\right) - y_{[i]}\right)^2$$

What about the entire dataset?

$$\boldsymbol{x} = \begin{bmatrix} x_{[1]} & x_{[2]} & \dots & x_{[n]} \end{bmatrix}^\top, \boldsymbol{y} = \begin{bmatrix} y_{[1]} & y_{[2]} & \dots & y_{[n]} \end{bmatrix}^\top$$

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \text{error}\left(f\left(x_{[i]}, \boldsymbol{\theta}\right), y_{[i]}\right) = \sum_{i=1}^{n} \left(f\left(x_{[i]}, \boldsymbol{\theta}\right) - y_{[i]}\right)^2$$

When $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ is small, then $f(x, \boldsymbol{\theta}) \approx y$ for the whole dataset!

Linear Regression - Optimization

Here is our loss function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \text{error}\left(f\left(x_{[i]}, \boldsymbol{\theta}\right), y_{[i]}\right) = \sum_{i=1}^{n} \left(f\left(x_{[i]}, \boldsymbol{\theta}\right) - y_{[i]}\right)^2$$

We want to find parameters $\boldsymbol{\theta}$ that make the loss small

We call this search for $\boldsymbol{\theta}$ **optimization**

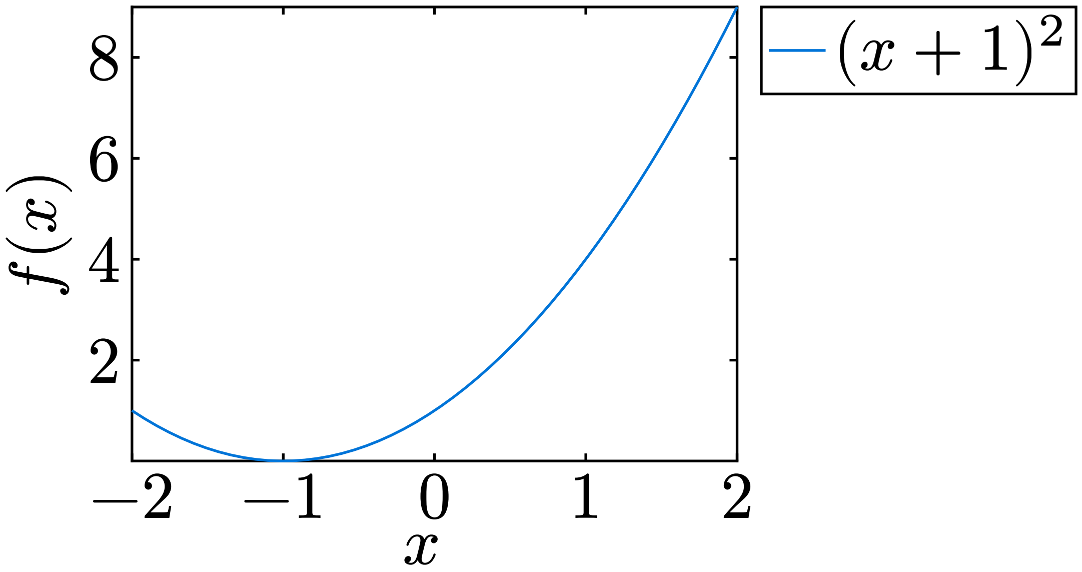
Let us state this more formally

Linear Regression - Optimization

Our objective is to **minimize** the loss, using arg min

$$\arg\min_x f(x)$$

Find x that makes $f(x)$ smallest



Question:

What is $\arg\min_x (x+1)^2$

Answer: $\arg\min_x (x+1)^2 = -1$, where $f(x) = 0$

Linear Regression - Optimization

Optimization objective: Find the $\boldsymbol{\theta}$ that minimizes the loss

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \operatorname{error}\left(f\left(x_{[i]}, \boldsymbol{\theta}\right), y_{[i]}\right)$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(f\left(x_{[i]}, \boldsymbol{\theta}\right) - y_{[i]}\right)^2$$

How do we optimize $\boldsymbol{\theta}$?

There is an analytical solution we will derive next lecture

For now, just follow these steps

Linear Regression - Optimization

First, we will construct a **design matrix** $\overline{\boldsymbol{X}}$ containing input data x

$$\overline{\boldsymbol{X}} = [\mathbf{1} \;\; \boldsymbol{x}] = \begin{bmatrix} 1 & x_{[1]} \\ 1 & x_{[2]} \\ \vdots & \vdots \\ 1 & x_{[n]} \end{bmatrix}$$

Question: Why two columns? **Hint:** How many parameters?

Linear Regression - Optimization

With our design matrix $\overline{\boldsymbol{X}}$ and desired output $\boldsymbol{y}$,

$$\overline{\boldsymbol{X}} = \begin{bmatrix} 1 & x_{[1]} \\ 1 & x_{[2]} \\ \vdots & \vdots \\ 1 & x_{[n]} \end{bmatrix}, \boldsymbol{y} = \begin{bmatrix} y_{[1]} \\ y_{[2]} \\ \vdots \\ y_{[n]} \end{bmatrix}$$

and our parameters $\boldsymbol{\theta}$,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix},$$

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{\theta}) = \left(\overline{\boldsymbol{X}}^\top \overline{\boldsymbol{X}}\right)^{-1} \overline{\boldsymbol{X}}^\top \boldsymbol{y}$$

Linear Regression - Optimization

To summarize:

The $\boldsymbol{\theta}$ given by

$$\arg\max_{\boldsymbol{\theta}} = \left(\overline{\boldsymbol{X}}^\top \overline{\boldsymbol{X}}\right)^{-1} \overline{\boldsymbol{X}}^\top \boldsymbol{y}$$

Provide the solution to

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \left(f\left(x_{[i]}, \boldsymbol{\theta}\right) - y_{[i]} \right)^2$$

Linear Regression - Optimization

Multiply $\overline{\boldsymbol{X}}$ and $\boldsymbol{\theta}$ to predict labels

$$f\left(x_{[1]}, \boldsymbol{\theta}\right)$$

$$\overline{\boldsymbol{X}}\boldsymbol{\theta} = \begin{bmatrix} 1 & x_{[1]} \\ 1 & x_{[2]} \\ \vdots & \vdots \\ 1 & x_{[n]} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \underbrace{\begin{bmatrix} \boxed{\theta_0 + \theta_1 x_{[1]}} \\ \theta_0 + \theta_1 x_{[2]} \\ \vdots \\ \theta_0 + \theta_1 x_{[n]} \end{bmatrix}}_{y \text{ prediction}}$$

We can also evaluate our model for new datapoints

$$\begin{bmatrix} 1 & x_{\text{Steven}} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \underbrace{\begin{bmatrix} \theta_0 + \theta_1 x_{\text{Steven}} \end{bmatrix}}_{y \text{ prediction}}$$

Linear Regression - Example Problem

Back to the example...

Task: Predict life expectancy

$X = \mathbb{R}_+$: Years in school

$Y = \mathbb{R}_+$: Age of death

Approach: Learn the parameters θ such that

$$f(x, \theta) = y; \quad x \in X, y \in Y$$

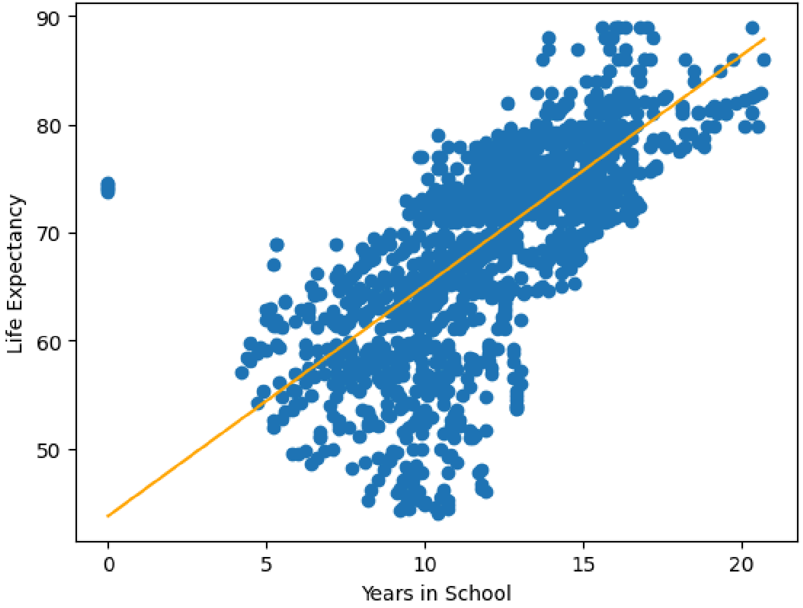
Goal: Given someone's education, predict how long they will live

You will be doing this in your first assignment!

Linear Regression - Example Problem

Plot the datapoints $\left(x_{[1]}, y_{[1]}\right), \left(x_{[2]}, y_{[2]}\right), \ldots$

Plot the curve $f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x; \quad x \in [0, 25]$

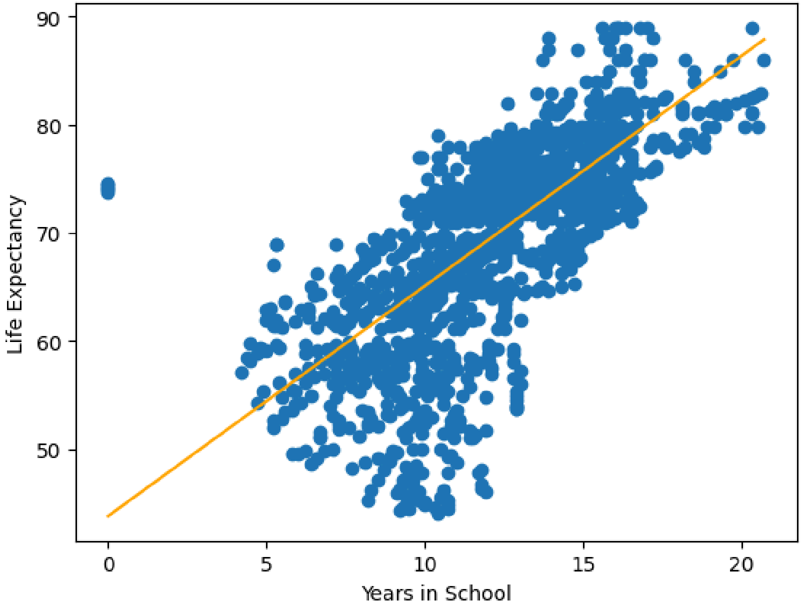


Linear Regression - Solve the Problem

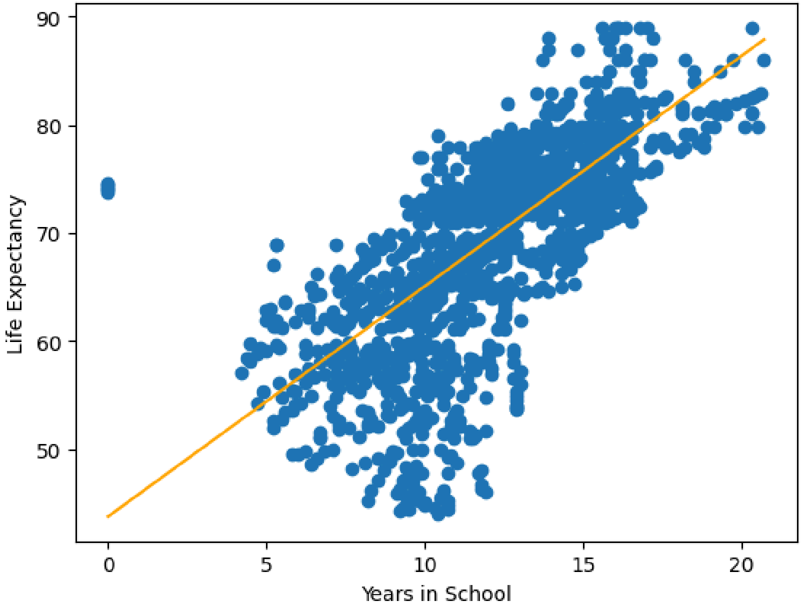
Goal: Given someone's education, predict how long they will live

Plot the datapoints $\left(x_{[1]}, y_{[1]}\right), \left(x_{[2]}, y_{[2]}\right), \ldots$

Plot the curve $f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x; \quad x \in [0, 25]$



Linear Regression - Solve the Problem



We figured out linear regression!

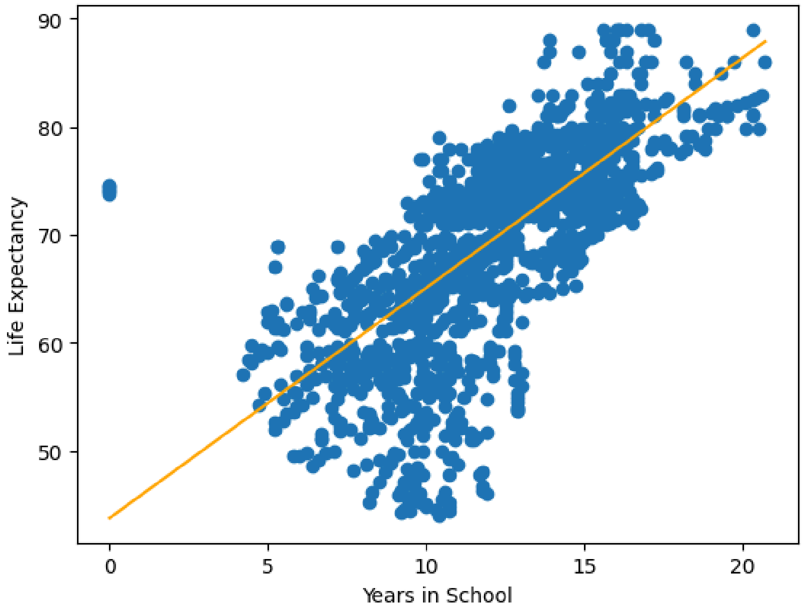
But can we do better?

Polynomial Regression

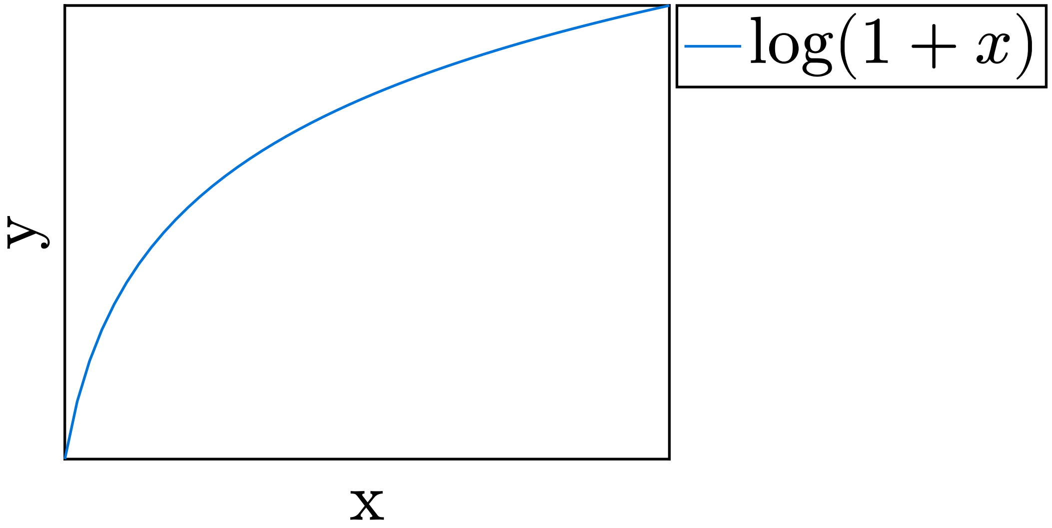
Polynomial Regression

Question:

Does the data look linear?



Or maybe more logarithmic?



However, linear regression must be linear!

Polynomial Regression

Question: What does it mean when we say linear regression is linear?

Answer: The function $f(x, \theta)$ is a linear function of x and θ

Trick: Change of variables to make f nonlinear: $x_{\text{new}} = \log(1 + x)$

$$\overline{\boldsymbol{X}} = \begin{bmatrix} 1 & x_{[1]} \\ 1 & x_{[2]} \\ \vdots & \vdots \\ 1 & x_{[n]} \end{bmatrix} \Rightarrow \overline{\boldsymbol{X}} = \begin{bmatrix} 1 & \log\left(1 + x_{[1]}\right) \\ 1 & \log\left(1 + x_{[2]}\right) \\ \vdots & \vdots \\ 1 & \log\left(1 + x_{[n]}\right) \end{bmatrix}$$

Now, f is a linear function of $\log(1 + x)$ – a nonlinear function of x!

Polynomial Regression

New design matrix...

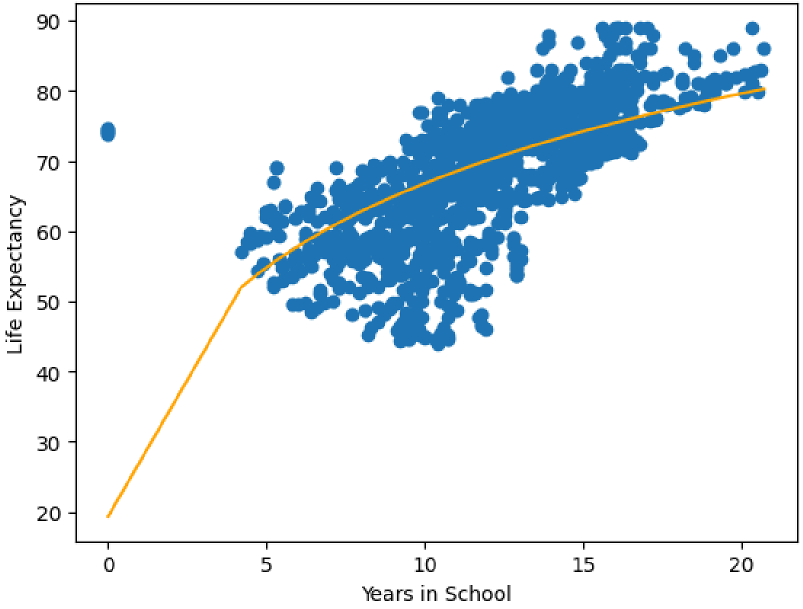
New **nonlinear** function...

$$\overline{\boldsymbol{X}} = \begin{bmatrix} 1 & \log\left(1 + x_{[1]}\right) \\ \vdots & \vdots \\ 1 & \log\left(1 + x_{[n]}\right) \end{bmatrix} \quad \overline{\boldsymbol{X}}\boldsymbol{\theta} = \begin{bmatrix} 1 & \log\left(1 + x_{[1]}\right) \\ \vdots & \vdots \\ 1 & \log\left(1 + x_{[n]}\right) \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \theta_0 + \theta_1 x_{[1]} \\ \vdots \\ \theta_0 + \theta_1 x_{[n]} \end{bmatrix}$$

Same solution...

$$\boldsymbol{\theta} = \left(\overline{\boldsymbol{X}}^\top \overline{\boldsymbol{X}}\right)^{-1} \overline{\boldsymbol{X}}^\top \boldsymbol{y}$$

Polynomial Regression



Better, but still not perfect

Can we do even better?

Polynomial Regression

What about polynomials?

$$f(x) = a + bx + cx^2 + \ldots + dx^m$$

Polynomials are **universal function approximators**

- Can approximate **any** function

Can we extend linear regression to polynomials?

Polynomial Regression

Expand x to a multi-dimensional input space...

$$\overline{\boldsymbol{X}} = \begin{bmatrix} 1 & x_{[1]} \\ 1 & x_{[2]} \\ \vdots & \vdots \\ 1 & x_{[n]} \end{bmatrix} \Rightarrow \overline{\boldsymbol{X}} = \begin{bmatrix} 1 & x_{[1]} & x_{[1]}^2 & \dots & x_{[1]}^m \\ 1 & x_{[2]} & x_{[2]}^2 & \dots & x_{[2]}^m \\ \vdots & \vdots & & \ddots & \\ 1 & x_{[n]} & x_{[n]}^2 & \dots & x_{[n]}^m \end{bmatrix}$$

Remember, n datapoints and $m + 1$ polynomial terms

And add some new parameters...

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_0 & \theta_1 \end{bmatrix}^\top \Rightarrow \boldsymbol{\theta} = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_m \end{bmatrix}^\top$$

Polynomial Regression

New function...

$$\overline{\boldsymbol{X}}\boldsymbol{\theta} = \underbrace{\begin{bmatrix} 1 & x_{[1]} & x_{[1]}^2 & \dots & x_{[1]}^m & 1 \\ 1 & x_{[2]} & x_{[2]}^2 & \dots & x_{[2]}^m & 1 \\ \vdots & \vdots & \ddots & & & \\ 1 & x_{[n]} & x_{[n]}^2 & \dots & x_{[n]}^m & 1 \end{bmatrix}}_{n \times (m+1)} \underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}}_{(m+1) \times 1} = \underbrace{\begin{bmatrix} \theta_0 + \theta_1 x_{[1]} + \dots + \theta_m x_{[1]}^m \\ \theta_0 + \theta_1 x_{[2]} + \dots + \theta_m x_{[2]}^m \\ \vdots \\ \theta_0 + \theta_1 x_{[n]} + \dots + \theta_m x_{[n]}^m \end{bmatrix}}_{\text{y prediction}, \ n \times 1}$$

Same solution... $\boldsymbol{\theta} = \left(\overline{\boldsymbol{X}}^\top \overline{\boldsymbol{X}}\right)^{-1} \overline{\boldsymbol{X}}^\top \boldsymbol{y}$

Polynomial Regression

$$f(x, \boldsymbol{\theta}) = f\left(x, \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} \right) = \theta_0 + \theta_1 x + ... + \theta_m x^m$$

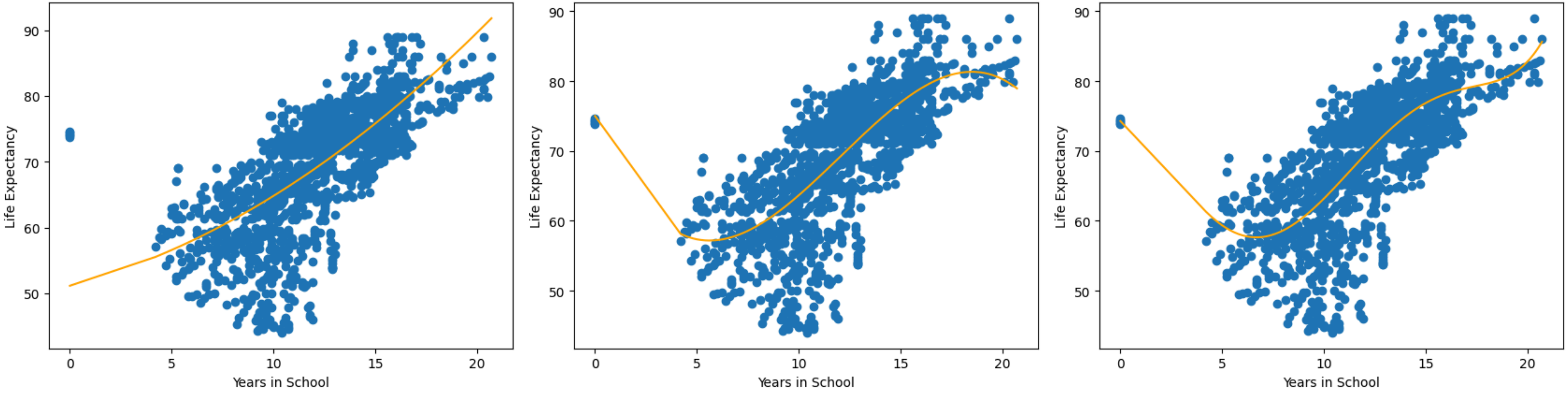
Summary: By changing the input space, we can fit a polynomial to the data using a linear fit!

Overfitting

Overfitting

$$f(x, \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \ldots + \theta_m x^m$$

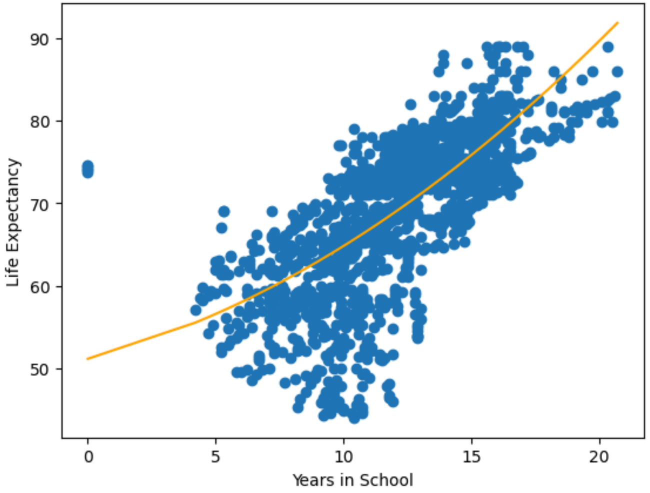
How do we choose m (polynomial order) that provides the best fit?



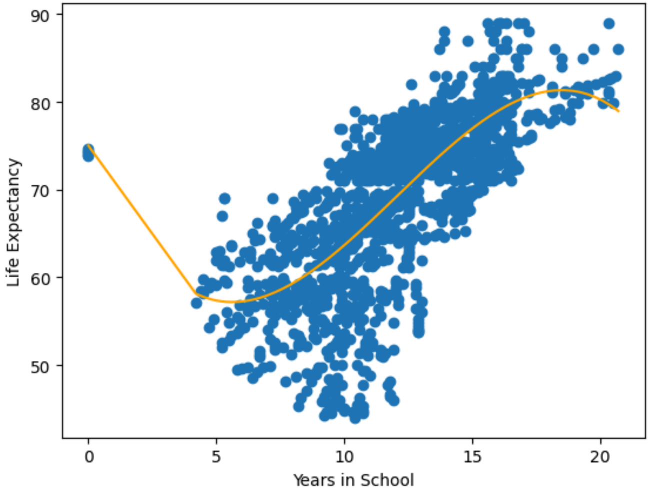
$m = 2$ $\quad$ $m = 3$ $\quad$ $m = 5$

Overfitting

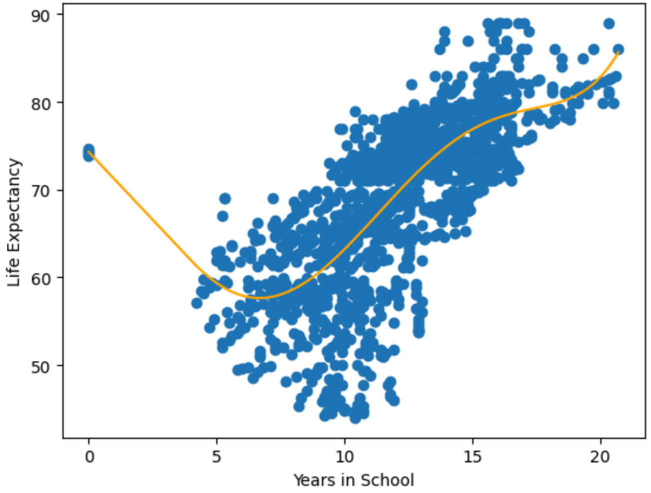
How do we choose n (polynomial order) that provides the best fit?



$$m = 2$$



$$m = 3$$

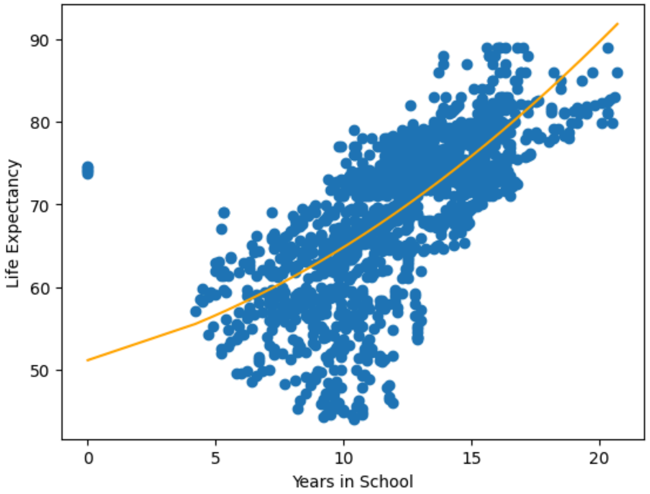


$$m = 5$$

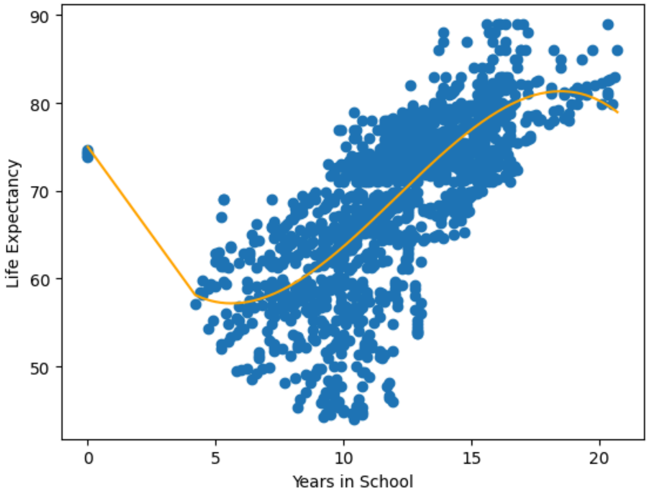
Pick the m with the smallest loss

$$\arg\min_{\boldsymbol{\theta}, m} \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}, (\boldsymbol{\theta}, m))$$

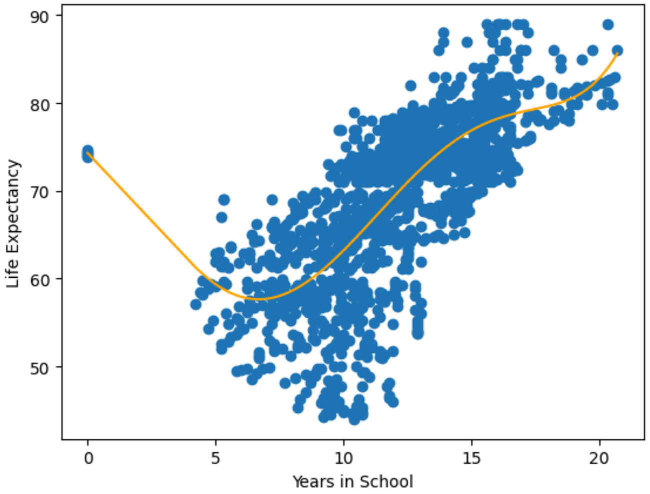
Overfitting



$m = 2$



$m = 3$

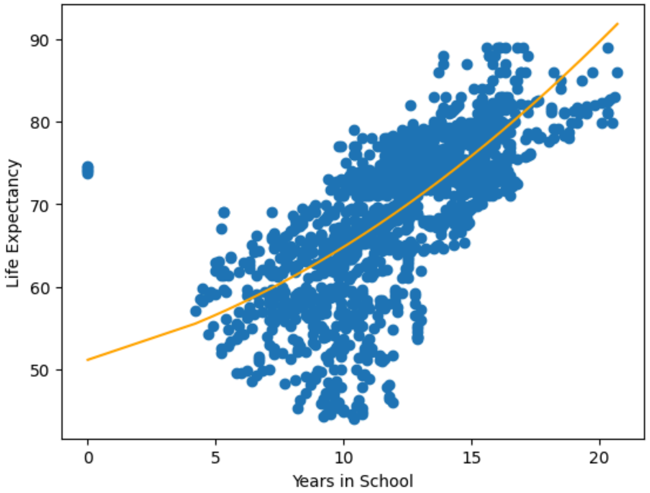


$m = 5$

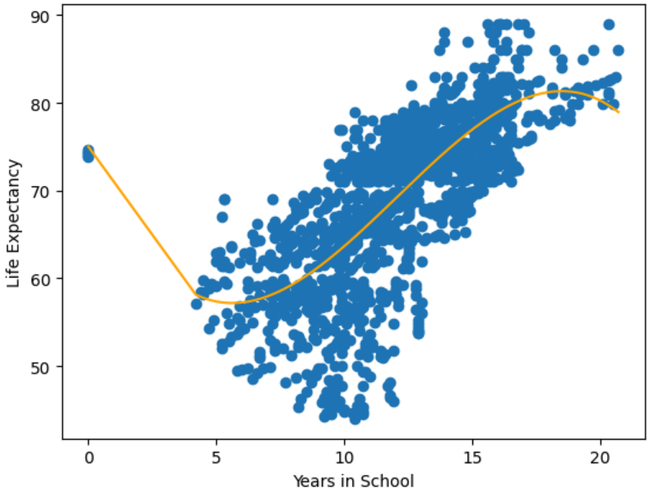
Question: Which m do you think has the smallest loss?

Answer: $m = 5$, but intuitively, $m = 5$ does not seem very good...

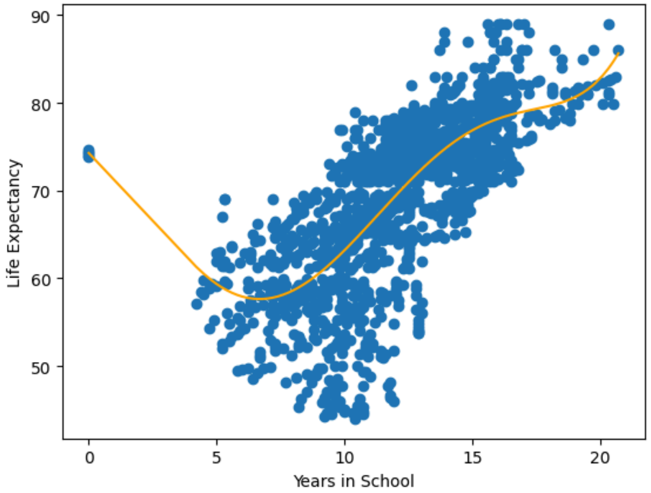
Overfitting



$m = 2$



$m = 3$



$m = 5$

More specifically, $m = 5$ will not generalize to new data

We will only use our model for new data (we already have the y for a known x)!

Overfitting

Model has a small loss but does not generalize to new data

We call this issue **overfitting**

The model fit too closely to data noise, rather than the trend

Models that overfit are not useful for making predictions

Back to the question...

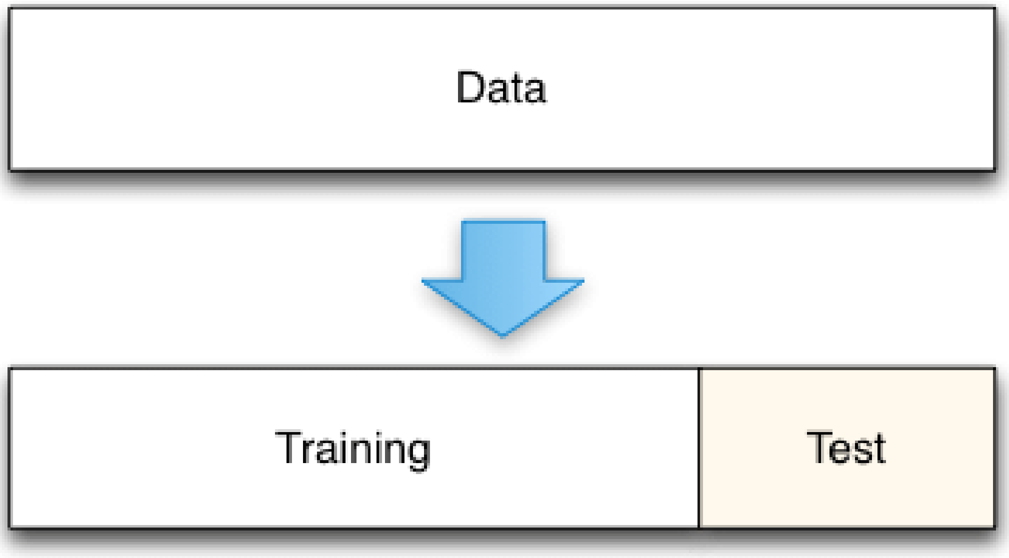
Question: How do we choose m such that our polynomial model works for unseen/new data?

Answer: Compute the loss on unseen data!

Overfitting

To compute the loss on unseen data, we will need unseen data

Let us create some unseen data!



Overfitting

Question: How do we choose the training and testing datasets?

$$\text{Option 1: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_{[1]} \\ x_{[2]} \\ x_{[3]} \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_{[1]} \\ y_{[2]} \\ y_{[3]} \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_{[4]} \\ x_{[5]} \end{bmatrix} \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_{[4]} \\ y_{[5]} \end{bmatrix}$$

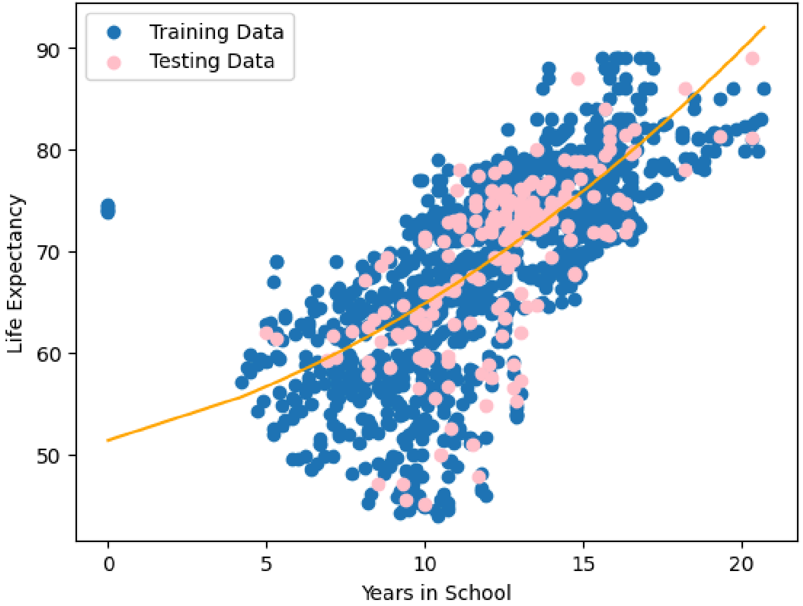
$$\text{Option 2: } \boldsymbol{x}_{\text{train}} = \begin{bmatrix} x_{[4]} \\ x_{[1]} \\ x_{[3]} \end{bmatrix} \boldsymbol{y}_{\text{train}} = \begin{bmatrix} y_{[4]} \\ y_{[1]} \\ y_{[3]} \end{bmatrix}; \quad \boldsymbol{x}_{\text{test}} = \begin{bmatrix} x_{[2]} \\ x_{[5]} \end{bmatrix} \boldsymbol{y}_{\text{test}} = \begin{bmatrix} y_{[2]} \\ y_{[5]} \end{bmatrix}$$

Answer: Always shuffle the data

Note: The model must never see the testing dataset during training. This is very important!

Overfitting

We can now measure how the model generalizes to new data



Learn parameters from the train dataset, evaluate on the test dataset

$$\mathcal{L}(\boldsymbol{X}_{\text{train}}, \boldsymbol{y}_{\text{train}}, \boldsymbol{\theta}) \qquad \mathcal{L}(\boldsymbol{X}_{\text{test}}, \boldsymbol{y}_{\text{test}}, \boldsymbol{\theta})$$

Overfitting

We use separate training and testing datasets on **all** machine learning models, not just linear regression

Homework

Homework

Homework 1 is released, due in two weeks

You will predict life expectancy based on education

- Maybe this convinces you to do a PhD

Homework

Tips for assignment 1

```python
def f(theta, design):
  # Linear function
  return design @ theta
```

Not all matrices can be inverted! Ensure the matrices are square and the condition number is low

```python
A.shape
cond = jax.numpy.linalg.cond(A)
```

Everything you need is in the lecture notes

Homework

https://colab.research.google.com/drive/1I6YgapkfaU71RdOotaTPLYdX9WflV1me