

# Lecture 12 – Multiple Linear Regression and Feature Engineering



DSC 40A, Spring 2023

# Announcements

- ▶ Homework 4 is out, due Tuesday at 11:59pm.
  - ▶ Assign pages to problems for full credit.
- ▶ Midterm 1 is **next Friday during lecture**.
  - ▶ Next Wednesday 7-9pm will be a mock exam and review session - save the date! No groupwork next week.
  - ▶ **Formula sheet** will be provided for the exam. No other notes.
  - ▶ More details coming soon.

# Agenda

- ▶ Incorporating multiple features.
- ▶ Interpreting parameters.
- ▶ Feature engineering.

**Incorporating multiple features**

## Last time

- ▶ We minimized the mean squared error for the prediction rule  $H(x) = w_0 + w_1x$ , which was

$$R_{sq}(\vec{w}) = \frac{1}{n} ||\vec{y} - X\vec{w}||^2$$

- ▶ We found that the minimizing  $\vec{w}$  satisfies the **normal equations**,  $X^T X \vec{w} = X^T \vec{y}$ .
  - ▶ If  $X^T X$  is invertible, the solution is:

$$\vec{w}^* = (X^T X)^{-1} X^T \vec{y}$$

- ▶ These same normal equations can be used to solve the **multiple linear regression** problem, where we use multiple features to predict an outcome. We simply need to adjust the design matrix  $X$ .

## Multiple linear regression example

- ▶ We want to fit a **linear** prediction rule with two features:

$$H(\text{experience}, \text{GPA}) = w_0 + w_1(\text{experience}) + w_2(\text{GPA})$$

- ▶ Collect data for each of  $n$  people:

Person #	Experience	GPA	Salary
1	3	3.7	85,000
2	6	3.3	95,000
3	10	3.1	105,000

- ▶ We represent each person with a **feature vector**:

$$\vec{x}_1 = \begin{bmatrix} 3 \\ 3.7 \end{bmatrix}, \quad \vec{x}_2 = \begin{bmatrix} 6 \\ 3.3 \end{bmatrix}, \quad \vec{x}_3 = \begin{bmatrix} 10 \\ 3.1 \end{bmatrix}$$

# Prediction rule form determines design matrix

- ▶ When our prediction rule is

$$H(\text{experience}, \text{GPA}) = \underline{w_0 + w_1(\text{experience}) + w_2(\text{GPA})},$$

the hypothesis vector  $\vec{h} \in \mathbb{R}^n$  can be written

$$\vec{h} = \begin{bmatrix} H(\text{experience}_1, \text{GPA}_1) \\ H(\text{experience}_2, \text{GPA}_2) \\ \dots \\ H(\text{experience}_n, \text{GPA}_n) \end{bmatrix} = \begin{bmatrix} 1 & \text{experience}_1 & \text{GPA}_1 \\ 1 & \text{experience}_2 & \text{GPA}_2 \\ \dots & \dots & \dots \\ 1 & \text{experience}_n & \text{GPA}_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

- ▶ Notice that the rows of the design matrix are the (transposed) feature vectors, with an additional 1 in front.

augmented

# Notation for multiple linear regression

- ▶ We will need to keep track of multiple<sup>1</sup> features for every individual in our data set.
- ▶ As before, subscripts distinguish between individuals in our data set. We have  $n$  individuals (or **training examples**).
- ▶ Superscripts distinguish between features.<sup>2</sup> We have  $d$  features.
  - ▶ experience =  $x^{(1)}$
  - ▶ GPA =  $x^{(2)}$

---

<sup>1</sup>In practice, we might use hundreds or even thousands of features.

<sup>2</sup>Think of them as new variable names, such as new letters.



# Augmented feature vectors

- The **augmented feature vector**  $\text{Aug}(\vec{x})$  is the vector obtained by adding a 1 to the front of feature vector  $\vec{x}$ :

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{bmatrix} \quad \text{Aug}(\vec{x}) = \begin{bmatrix} 1 \\ x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{bmatrix} \quad \vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

- Then, our prediction rule is

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1 x^{(1)} + w_2 x^{(2)} + \dots + w_d x^{(d)} \\ &= \vec{w} \cdot \text{Aug}(\vec{x}) \end{aligned}$$

# The general problem

- We have  $n$  data points (or **training examples**):  
 $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$  where each  $\vec{x}_i$  is a feature vector of  $d$  features:

$$\vec{x}_i = \begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \\ \dots \\ x_i^{(d)} \end{bmatrix}$$

person  $i$   
2<sup>nd</sup> piece  
of info  
(GPA)

- We want to find a good linear prediction rule:

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1 x^{(1)} + w_2 x^{(2)} + \dots + w_d x^{(d)} \\ &= \vec{w} \cdot \text{Aug}(\vec{x}) \end{aligned}$$

## The general solution

- Use design matrix

*row is a person*

*col is a feature*

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(d)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(d)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(d)} \end{bmatrix} = \begin{bmatrix} \text{Aug}(\vec{x}_1)^T \\ \text{Aug}(\vec{x}_2)^T \\ \dots \\ \text{Aug}(\vec{x}_n)^T \end{bmatrix}$$

and observation vector to solve the **normal equations**

$$X^T X \vec{w}^* = X^T \vec{y}$$

to find the optimal parameter vector.

## Terminology for parameters

- ▶ With  $d$  features,  $\vec{w}$  has  $d + 1$  entries.

→ intercept plus one for each feature

- ▶  $w_0$  is the bias, also known as the intercept.

- ▶  $w_1, \dots, w_d$  each give the weight, i.e. coefficient, of a feature.

$$H(\vec{x}) = \underline{w_0} + \underline{w_1}x^{(1)} + \dots + \underline{w_d}x^{(d)}$$

parameters

## Interpreting parameters

## Example: predicting sales

- ▶ For each of 26 stores, we have:

- ▶ net sales,
- ▶ square feet,
- ▶ inventory,
- ▶ advertising expenditure,
- ▶ district size, and
- ▶ number of competing stores.

- ▶ Goal: predict net sales given these features

- ▶ To begin:

$$H(\text{square feet}, \text{competitors}) = w_0 + w_1(\text{square feet}) + w_2(\text{competitors})$$

## Example: predicting sales

sales

$$H(\text{square feet, competitors}) = w_0 + w_1(\text{square feet}) + w_2(\text{competitors})$$

### Discussion Question

What will be the sign of  $w_1^*$  and  $w_2^*$ ?

- a)  $w_1^* = +$ ,  $w_2^* = -$
- b)  $w_1^* = +$ ,  $w_2^* = +$
- c)  $w_1^* = -$ ,  $w_2^* = -$
- d)  $w_1^* = -$ ,  $w_2^* = +$

## Example: predicting sales

$$H(\text{square feet, competitors}) = w_0 + w_1(\text{square feet}) + w_2(\text{competitors})$$

### Discussion Question

What will be the sign of  $w_1^*$  and  $w_2^*$ ?

- a)  $w_1^* = +$ ,  $w_2^* = -$
- b)  $w_1^* = +$ ,  $w_2^* = +$
- c)  $w_1^* = -$ ,  $w_2^* = -$
- d)  $w_1^* = -$ ,  $w_2^* = +$

Let's try it out ourselves. [Follow along here.](#)



## Which features are most “important”?

### Discussion Question

Which feature has the greatest effect on the outcome?

- a) square feet:  $w_1^* = 16.202$
- b) competitors:  $w_2^* = -5.311$
- c) inventory:  $w_2^* = \underline{0.175}$
- d) advertising:  $w_3^* = 11.526$
- e) district size:  $w_4^* = 13.580$

## Which features are most “important”?

- ▶ The most important feature is **not necessarily** the feature with largest weight.
- ▶ Features are measured in different units, scales.
  - ▶ Suppose I fit one prediction rule,  $H_1$ , with sales in dollars, and another prediction rule,  $H_2$ , with sales in thousands of dollars.
  - ▶ Sales is just as important in both prediction rules.
  - ▶ But the weight of sales in  $H_1$  will be 1000 times smaller than the weight of sales in  $H_2$ .
  - ▶ Intuitive explanation:  $5 \times 45000 = (5 \times 1000) \times 45$ .
- ▶ **Solution:** before doing regression, standardize each feature, i.e. convert each feature to standard units.

## Standard units

- Recall: to convert a feature  $x_1, x_2, \dots, x_n$  to standard units, we use the formula

$$x_i \text{ in standard units} = \frac{x_i - \bar{x}}{\sigma_x}$$

$\bar{x}$  → mean  
 $\sigma_x$  → standard deviation

- Example: 1, 7, 7, 9

Mean:  $24/4 = 6$

Standard deviation:

$$\text{var} = \frac{1}{4} (5^2 + 1^2 + 1^2 + 3^2)$$

$$\text{var} = \frac{1}{4} \cdot 36 = 9 \Rightarrow \text{SD} = \sqrt{9} = 3$$

Standardized data:

$$1 \Rightarrow \frac{1-6}{3} = -\frac{5}{3} \text{ s.u.}$$

$$7 \Rightarrow \frac{7-6}{3} = \frac{1}{3} \text{ s.u.}$$

$$9 \Rightarrow \frac{9-6}{3} = 1 \text{ s.u.}$$



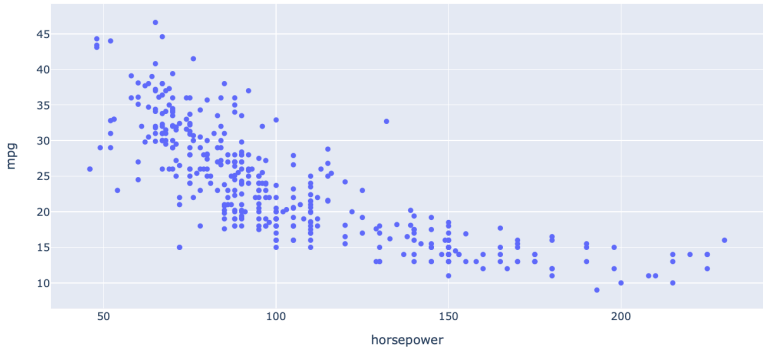
# Standard units for multiple linear regression

- ▶ The result of standardizing each feature (separately!) is that the units of each feature are on the same scale.
  - ▶ There's no need to standardize the outcome (net sales), since it's not being compared to anything.
- ▶ Then, solve the normal equations. The resulting  $w_0^*, w_1^*, \dots, w_d^*$  are called the standardized regression coefficients.
- ▶ Standardized regression coefficients can be directly compared to one another.

Let's try it out in our demo notebook.

## Feature engineering

MPG vs. Horsepower



**Question:** Would a linear prediction rule work well on this dataset?

## A quadratic prediction rule

- ▶ It looks like there's some sort of quadratic relationship between horsepower and mpg in the last scatter plot. We want to try and fit a prediction rule of the form

$$H(x) = w_0 + w_1 x + w_2 x^2$$

- ▶ Note that while this is quadratic in horsepower, it is **linear in the parameters!**
- ▶ We can do that, by choosing our two “features” to be  $x_i$  and  $x_i^2$ , respectively.
  - ▶ In other words,  $x_i^{(1)} = x_i$  and  $x_i^{(2)} = x_i^2$ .
  - ▶ More generally, we can create new features out of existing features.

## A quadratic prediction rule

- ▶ Desired prediction rule:  $H(x) = w_0 + w_1x + w_2x^2$ .
- ▶ The resulting design matrix looks like this:

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \dots & & \\ 1 & x_n & x_n^2 \end{bmatrix}$$

- ▶ To find optimal parameter vector  $\vec{w}^*$ : solve the **normal equations**!

$$X^T X w^* = X^T y$$



## More examples

- ▶ What if we want to use a prediction rule of the form  $H(x) = w_0 + w_1x + w_2x^2 + w_3x^3$ ?
- ▶ What if we want to use a prediction rule of the form  $H(x) = w_1\frac{1}{x^2} + w_2 \sin x + w_3e^x$ ?

# Feature engineering

- ▶ More generally, we can create new features out of existing information in our dataset. This process is called **feature engineering**.
  - ▶ In this class, feature engineering will mostly be restricted to creating non-linear functions of existing features (as in the previous example).
  - ▶ In the future you'll learn how to do other things, like encode categorical information.

## Summary

## Summary

- ▶ The normal equations can be used to solve the **multiple linear regression** problem, where we use multiple features to predict an outcome.
- ▶ We can interpret the parameters as weights. The signs of weights give meaningful information, but we can only compare weights if our features are standardized.
- ▶ We can create non-linear features out of existing features. This process is called **feature engineering**.
  - ▶ A prediction rule only needs to be a **linear function of the parameters** for us to use linear regression. It does not need to be a linear function of the features.