DSC 40A

Theoretical Foundations of Data Science I

Last Time

- We found that any prediction rule that was **linear in the parameters** could be solved by the **normal equations**

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

In This Video

We will make predictions based on multiple features and interpret the resulting prediction rules.

Recommended Reading

Course Notes: Chapter 2, Section 2
Review: Linear Algebra Textbook

Using Multiple Features

- How do we predict salary given **multiple** features?
- We believe salary is a function of experience *and* GPA.
- I.e., there is a function H so that:

$$\text{salary} \approx H(\text{years of experience}, \text{GPA})$$

- Recall: H is a **prediction rule**.
- **Our goal**: find a good prediction rule, H.

Example Prediction Rules

$$H_1(\text{experience}, \text{GPA}) = \$2,000 \times (\text{experience}) + \$40{,}000 \times \frac{\text{GPA}}{4.0}$$

$$H_2(\text{experience}, \text{GPA}) = \$60{,}000 \times 1.05^{(\text{experience}+\text{GPA})}$$

$$H_3(\text{experience}, \text{GPA}) = \cos(\text{experience}) + \sin(\text{GPA})$$

Linear Prediction Rule

- We'll restrict ourselves to **linear** prediction rules:

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

- This is called **multiple linear regression.**

- Since H is **linear in the parameters** w_0, w_1, w_2, the solution comes from solving the **normal equations**.

The Data

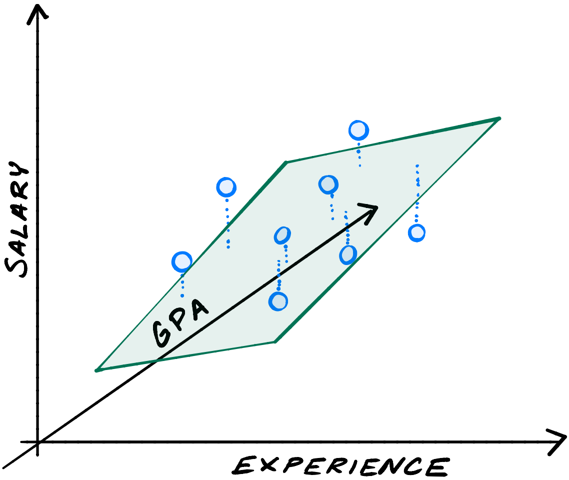
- For each of n people, collect each feature, plus salary:

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**:

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

Geometric Interpretation



The Hypothesis Vector

- When our prediction rule is

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

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

$$\vec{h} = \begin{bmatrix} H(\text{experience}_1, \text{GPA}_1) \\ H(\text{experience}_2, \text{GPA}_2) \\ \vdots \\ H(\text{experience}_n, \text{GPA}_n) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \text{experience}_1 & \text{GPA}_1 \\ 1 & \text{experience}_2 & \text{GPA}_2 \\ \vdots & \vdots & \vdots \\ 1 & \text{experience}_n & \text{GPA}_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}.$$

Solution

- Use design matrix

$$X = \begin{bmatrix} 1 & \text{experience}_1 & \text{GPA}_1 \\ 1 & \text{experience}_2 & \text{GPA}_2 \\ \vdots & \vdots & \vdots \\ 1 & \text{experience}_n & \text{GPA}_n \end{bmatrix}$$

  and solve the **normal equations**

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

  to find the optimal choice of parameters.

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

Notation for Multiple Linear Regression

- We will need to keep track of multiple[1] 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.[2] We have d features.
  - experience $= x^{(1)}$
  - GPA $= x^{(2)}$

[1] In practice, might use hundreds or even thousands of features.
[2] 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 $\vec{x}$:

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

- Then, our prediction rule is

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

The General Problem

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

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

- We want to find a good linear prediction rule:

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

The General Solution

- Use design matrix

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(d)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(d)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(d)} \end{bmatrix} = \begin{bmatrix} Aug(\vec{x_1})^T \\ Aug(\vec{x_2})^T \\ \vdots \\ Aug(\vec{x_n})^T \end{bmatrix}$$

and solve the **normal equations**

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

to find the optimal choice of parameters.

Interpreting the Parameters

- With d features, $\vec{w}$ has $d+1$ entries.
- w_0 is the **bias**.
- $w_1, \ldots, w_d$ each give the **weight** of a feature.

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

- Sign of w_i tells us about relationship between ith feature and outcome.

Example: Predicting Sales

- For each of 26 stores, we have:
  - net sales,
  - size (sq ft),
  - inventory,
  - advertising expenditure,
  - district size,
  - number of competing stores.

- Goal: predict net sales given size, inventory, etc.

- To begin:

$$H(\text{size}, \text{competitors}) = w_0 + w_1 \times \text{size} + w_2 \times \text{competitors}$$

Example: Predicting Sales

$$H(\text{size}, \text{competitors}) = w_0 + w_1 \times \text{size} + w_2 \times \text{competitors}$$

Question

What will be the sign of w_1 and w_2?

A) $w_1 = +, \quad w_2 = -$
B) $w_1 = +, \quad w_2 = +$
C) $w_1 = -, \quad w_2 = -$
D) $w_1 = -, \quad w_2 = +$

Demo

Question

Which feature has the greatest effect on the outcome?

A) size:	$w_1 = 16.20$
B) inventory:	$w_2 = 0.17$
C) advertising:	$w_3 = 11.53$
D) district size:	$w_4 = 13.58$
E) competing stores:	$w_5 = -5.31$

Which features are most “important”?

- **Not necessarily** the feature with largest weight.
- Features are measured in different units, scales.
- We should **standardize** each feature.

Standard Units

- To standardize (z-score) a feature, subtract mean, divide by standard deviation.

- Example: 1, 7, 7, 9
  - Mean: 6
  - Standard Deviation:

$$\sqrt{\frac{1}{4}((-5)^2 + (1)^2 + (1)^2 + (3)^2)} = 3$$

  - Standardized Data:

$$\frac{1-6}{3} = -\frac{5}{3}, \qquad \frac{7-6}{3} = \frac{1}{3}, \qquad \frac{7-6}{3} = \frac{1}{3}, \qquad \frac{9-6}{3} = 1$$

- Measures number of standard deviations *above* the mean.

Standard Units for Multiple Regression

- Standardize each feature (store size, inventory, etc.) separately.
- No need to standardize outcome (net sales).
- Solve normal equations. The resulting $w_0, w_1, \ldots, w_d$ are called the **standardized regression coefficients**.
- They can be directly compared to one another.

Demo

Nonlinear Function of Multiple Features

- Suppose we want to fit a rule of the form:

$$\begin{aligned} H(\text{size}, \text{competitors}) &= w_0 + w_1 \text{size} + w_2 \text{size}^2 \\ &\qquad + w_3 \text{competitors} + w_4 \text{competitors}^2 \\ &= w_0 + w_1 s + w_2 s^2 + w_3 c + w_4 c^2 \end{aligned}$$

- Make design matrix:

$$X = \begin{bmatrix} 1 & s_1 & s_1^2 & c_1 & c_1^2 \\ 1 & s_2 & s_2^2 & c_2 & c_2^2 \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & s_n & s_n^2 & c_n & c_n^2 \end{bmatrix}$$

Where c_i and s_i are the competitors and size of the ith store.

Summary

- The normal equations can be used to solve the **multiple linear regression** problem.

- Interpret the parameters as weights. Signs give meaningful information, but only compare weights if data is standardized.