Linear Regression I

COSC 410: Applied Machine Learning
Fall 2025
Prof. Forrest Davis

September 11, 2025

Warm-up

1. Talk to your neighbor about your favorite dessert
2. Given the data in Figure 1, what is the Gini impurity of the root node (before we apply a decision boundary)?

Logistics

- Codelet 1 is due Friday
- Lab 2 is due Friday
- Codelet 2 is posted on the website
  - Implementing linear regression based on the book

Learning Objectives

- Describe the basic aims of linear regression
- Map the linear regression problem to equation
- Apply some key concepts from linear regression
- Apply a regression loss
- Describe the optimization problem for linear regression

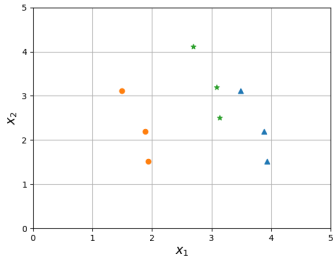

Figure 1: A dataset with two features and three output labels (an orange circle, a blue triangle, and a green star).

Summary: We lay out the motivation, formalization, and loss for linear regression. Along the way, we refresh some useful mathematical operations and start building intuitions about representational learning.

Model Goal

LINEAR REGRESSION WILL BE THE BASIS, surprisingly, for vanilla neural networks, with some additional small tweaks. So let's dig into this type of model a bit more deeply and see if we can build solid

intuitions that will help us later. Let's start with some fake data, in Figure 2.

At its core the aim of regression is to predict a numerical value drawn from some continuous distribution. In our example, we want to uncover the relationship between x_1, and y. Our **assumption** in linear regression is that this relationship is linear.

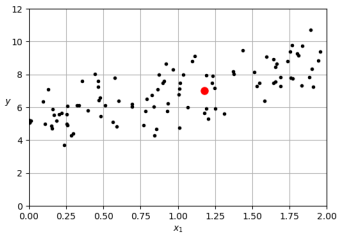

Figure 2: A basic dataset with one feature and a continuous output label. One sample data point is highlighted in red.

[1] The superscript here refers to the sample and the subscript refers to the dimension of the input. For example, $x_3^{(8)}$ refers to the 3rd dimension – 4th dimension if we count from zero – of the 8th – or 9th if we count from zero – input.

[2] That is, we are assuming the output is determined by a scaling of our input plus a term, which we call the *bias*.

Starting Small: Singly None

WE WILL BEGIN WITH A SINGLE POINT, highlighted in red in Figure 2. We are trying to find parameters that let's us predict the observed y value from the input, $x^{(0)}$.[1] In this case, we want to find parameters that maps from 1.18 (our sample's first feature value) to 7. In other words, we are trying to solve for w_1 in the following equation.[2]

$$y = w_1^{(1)} x_1^{(0)} + b$$

Question

What values would you put for w_1 and b and why?

Building Out: Many Stones Can Form a Line

IN CONSIDERING MANY SINGLE DIMENSIONAL SAMPLES, we need to draw on a mathematical object in our skill set, **vectors**. Recall, a vector is an array of scalars, for example, $\mathbf{a}$ is a 3 dimensional vector (i.e., $\mathbf{a} \in \mathbb{R}^3$):[3]

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

[3] In $\mathbf{a}$, each row is a sample. We have three samples each with one feature.

Suppose we considered all our points, we would have two vectors: $\mathbf{x}$ and $\mathbf{y}$. Then we are working with

$$\hat{\mathbf{y}} = w_1 \mathbf{x} + b$$

We can try out some different parameters and see which is better. We are comparing our model's predictions, which we call $\hat{\mathbf{y}}$ (pronounced, y hat), to our true outputs, $\mathbf{y}$ (sometimes called gold outputs).

The Foundation: One Sample with Many Features

WHAT IF OUR INPUT IS COMPOSED of more than one feature (e.g., houses represented by number of rooms, number of bathrooms, square footage, etc.)? We'd then want to learn a weight (or parameter) for each input feature:

$$y = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n + b$$

Notice that we have a number of scalar parameters and a number of scalar input features. Perhaps, we can write this with reference to two vectors, $\mathbf{w}$ and $\mathbf{x}$.

To do this, recall the **dot product**.[4] The dot product is defined between vectors of equal length n as:

$$\mathbf{v} \cdot \mathbf{w} = \sum_{i=0}^{n} \mathbf{v}_i\mathbf{w}_i = \mathbf{v}_0\mathbf{w}_0 + \mathbf{v}_1\mathbf{w}_1 + \cdots + \mathbf{v}_n\mathbf{w}_n$$

That is, the dot product of two vectors yields a scalar. We can interpret this scalar as a measure of the angle between the vectors. Formally, the dot product, in geometric terms, is:

$$\mathbf{v} \cdot \mathbf{w} = ||\mathbf{v}||||\mathbf{w}||\cos(\theta)$$

Where $||\mathbf{v}||$ is the magnitude (length) of the vector $\mathbf{v}$. The definition should be familiar if you recall euclidean distance:

$$||\mathbf{v}|| = \sqrt{v_0^2 + v_1^2 + \cdots + v_n^2}$$

We can visualize this relationship by noting that the dot product projects one vector onto the other as in Figure 3.

[4] For a refresher on dot products, see this video.

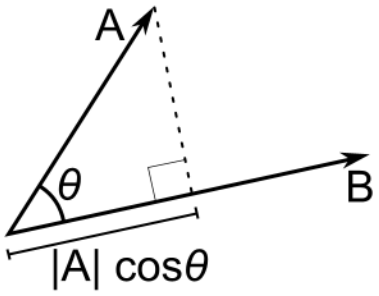

Figure 3: Projection of one vector onto another.
This applet is a nice way to interactively see the relationship between magnitude, direction, and the angle between vectors via the dot product.

Question

1. What is the dot product between $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$?

2. What are two non-zero vectors (zero vectors are vectors of all zeros) which have a dot product of zero with $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$?

We can now represent our model more succinctly using vectors as

$$\hat{y} = \mathbf{w} \cdot \mathbf{x} + b$$

We can make this even more succinct by add the bias to our weight vector $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \dots \\ w_n \\ b \end{bmatrix}$ and adding a 1 to the bottom of our input vector $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \\ 1 \end{bmatrix}$. Now,

$$\hat{y} = \mathbf{w} \cdot \mathbf{x}$$

The Building: Many Samples with Many Dimensions

FINALLY, WE SHOULD CONSIDER HOW TO REPRESENT MANY SAMPLES of many dimensions. To represent this, we will use **matrices**. Following our early convention with rows as samples and columns as features, we can translate our n-dimensional samples into a matrix like the following

$$\mathbf{X} = \begin{pmatrix} a_0^{(1)} & a_1^{(1)} & a_2^{(1)} & \dots & a_n^{(1)} \\ a_0^{(2)} & a_1^{(2)} & a_2^{(2)} & \dots & a_n^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ a_n^{(m)} & a_1^{(m)} & a_2^{(m)} & \dots & a_n^{(m)} \end{pmatrix}$$

We need some way of applying or parameters to these samples. Luckily we have matrix multiplication. Recall, our quickly learn, that matrix multiplication builds on the dot product as a means of multiplying matrices together.[5] It is visually depicted in Figure 4.

Formally, suppose we have two matrices:

$$\mathbf{A} = \begin{pmatrix} a_{(0,0)} & a_{(0,1)} & a_{(0,2)} \\ a_{(1,0)} & a_{(1,1)} & a_{(1,2)} \\ a_{(2,0)} & a_{(2,1)} & a_{(2,2)} \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} b_{(0,0)} & b_{(0,1)} & b_{(0,2)} \\ b_{(1,0)} & b_{(1,1)} & b_{(1,2)} \\ b_{(2,0)} & b_{(2,1)} & b_{(2,2)} \end{pmatrix}$$

Matrix multiplication would return a new matrix:

$$\mathbf{C} = \begin{pmatrix} c_{(0,0)} & c_{(0,1)} & c_{(0,2)} \\ c_{(1,0)} & c_{(1,1)} & c_{(1,2)} \\ c_{(2,0)} & c_{(2,1)} & c_{(2,2)} \end{pmatrix}$$

where:

[5] See this video series for a refresher on matrix multiplication.

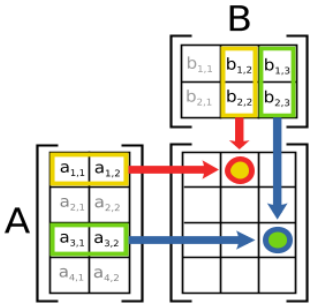

Figure 4: Visual representation of matrix multiplication to determine the output in two cells.

$$
\begin{aligned}
c_{(i,j)} &= \sum_{k=0}^{2} a_{(i,k)} b_{(k,j)} \\
&= a_{(i,0)} b_{(0,j)} + a_{(i,1)} b_{(1,j)} + a_{(i,2)} b_{(2,j)}
\end{aligned}
$$

Question

1. What is $\mathbf{AB}$ when $\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 3 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 2 & 1 \\ 0 & 3 \\ 1 & -1 \end{bmatrix}$?

2. Matrix multiplication has a shape restriction. What is it?

Question

How do we generalize our linear regression equation to work with matrices?

Notice, that matrix multiplication is nothing more than the dot product between the rows in $\mathbf{A}$ and the columns in $\mathbf{B}$. So we can use matrix multiplication as a way of doing many dot products between vectors.

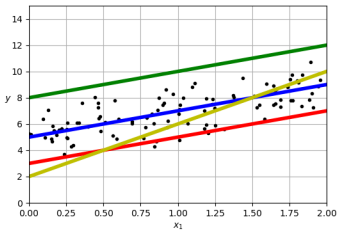

Figure 5: Four possible models fit to our sample data.

Loss Function

CONSIDER THE POSSIBLE MODELS IN Figure 5. Which is better and why? One natural option to quantify goodness is to say that a line is a good fit to our data if it is as close as possible to our data. That is, we will consider are predictions $\hat{\mathbf{y}}$ and our true labels $\mathbf{y}$. We can ask, how close were we with each prediction? We quantify this as $y^{(i)} - \hat{y}^{(i)}$. We need some way of aggregating over all our predictions, one good option is the mean:

$$\frac{1}{m} \sum_{i=0}^{m-1} y^{(i)} - \hat{y}^{(i)}$$

Practice Problems

We are given true targets y and predictions $\hat{y}$:

sample	y	$\hat{y}$
1	3	2
2	-1	0
3	5	4
4	2	3

What is the loss associated with this model?

Question

What is a problem with our current loss function based on what you just calculated?

The Land: Learning as Optimization

GIVEN A DATASET, A MODEL, AND A LOSS, we would like to use these productively to solve a task. With parameterized models, like linear regression, we are after the set of parameters that *minimize* our loss (i.e., that are good). In standard linear regression, we are seeking to minimize **mean squared error** over our dataset.[6]

$$\arg\min_{\mathbf{w}\in\mathbb{R}^{n+1}} \text{MSE}(\mathbf{X}, \mathbf{y}; \mathbf{w})$$

We can visualize our optimization goal by comparing how different parameters ($\mathbf{w}$'s in our case) impact the MSE over all of our data. Figure 6 does just that.

Question

Based on Figure 6, what parameters do we want and why?

Before Next Class

- Pre-class quiz
- Work on codelets

[6] $\mathbf{w}$ is an element of $\mathbb{R}^{n+1}$ and not $\mathbb{R}^n$ because we are folding in the bias term, so there are n parameters for each feature of our input and 1 bias term.

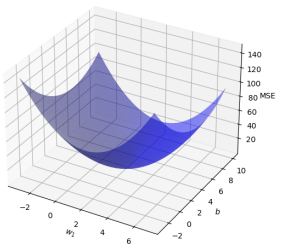

Figure 6: Mean-squared error with different parameters.