CS 337, Fall 2023
Introduction to Linear Regression

Scribes: Kartik S. Nair*, Ravindra Mohith*, Ravi Shankar Meena*,
Palle Bhavana, Vaibhav Vishal, Gowtham S
Edited by: Ashish Agrawal

3 August 2023

Disclaimer. Please note this document has not received the usual scrutiny that formal publications enjoy. This may be distributed outside this class only with the permission of the instructor.

1 Notation & Introduction

Before we introduce the Linear Regression problem, let us introduce some of the basic notation and terminology used.

In general we format variables as x for scalars, $\mathbf{x}$ for vectors, and X for matrices.

- Input / *Feature space* / Attributes Space : $\mathcal{X} = \mathbb{R}^d$ for some $d \in \mathbb{N}$
- Output / *Label space* / Response space : $\mathcal{Y} = \mathbb{R}$
- Dataset : $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y} \;\; \forall i \in \{1 \dots n\}$

Consider a target function $f : \mathcal{X} \to \mathcal{Y}$ on the training dataset $\mathcal{D}$, i.e. $\mathbf{x} \overset{f}{\mapsto} y \;\; \forall \; (\mathbf{x}, y) \in \mathcal{D}$.
The focus is to find a *hypothesis function* $h : \mathcal{X} \to \mathcal{Y}$ that ideally closely approximates f. We call the family of the hypothesis functions as $\mathcal{H}$, the *Hypothesis Class*. This now brings the following questions:

1. What are the possibilities for the predictor function h ? [**Hypothesis Class**]
2. How do you quantify the performance of the predictor? [**Loss/Error Function**]
3. How do we find the best predictor? [**Optimization**]

2 What are the possible predictor functions?

Let us first play with a simpler case with a one-dimensional feature space. We may consider the problem as a line fitting problem, taking our hypothesis class to be all linear functions.

Let us parametrize the line with w_0, w_1 (intercept and slope). We can vectorize our parameters as $W = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$.

Then we may write our hypothesis function $h_{\mathbf{w}}$ parameterised by $\mathbf{w}$ as.

$$h_{\mathbf{w}}(x) = w_0 + w_1 x$$

*Larger credit to these scribes for the final notes.

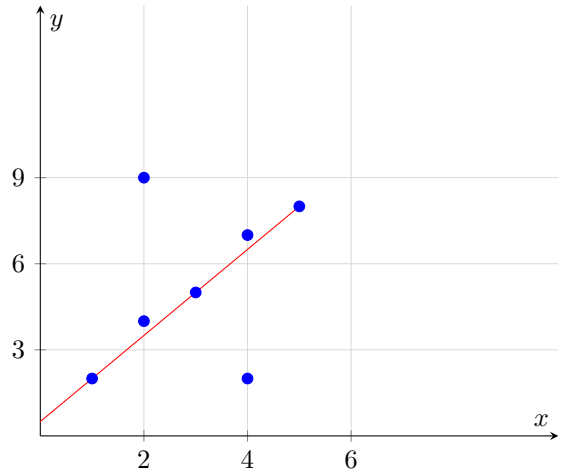

Figure 1: Fitting a line to the data

Let us also vectorize our input as $\mathbf{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}$, so that

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$$

We can now extend this to the multi-dimensional case. We consider our function to return a linear combination of d dimensional features.

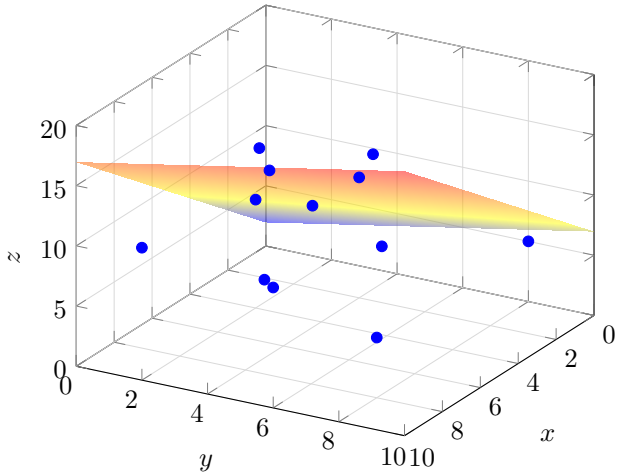

Figure 2: For higher dimensional feature spaces, linear regression is akin to hyperplane fitting

We now have our parameter $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \in \mathbb{R}^{d+1}$ and input vector $\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$ to get an identical expression:

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$$

Our hypothesis class is, then:

$$\mathcal{H} = \{h_{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^{d+1}\}$$

This is essentially the *linear* in linear regression. However, note that it does not mean we are restricted to linear functions of the features, we may transform the feature space to another space to regress.

3 Quantifying the performance of a predictor

We define a function that operates on the predictor function and dataset to quantify the "mismatch" between the two. Higher the loss function, lesser is the given predictor suitable for the dataset. Given that our function is parameterized, we may also define the loss function in terms of the parameter.

$$\text{Loss Function: } \mathcal{L}(h, \mathcal{D}) \text{ or } \mathcal{L}(\mathbf{w}, \mathcal{D})$$

Let us consider a singleton dataset $\mathcal{D}_{test} = \{(\mathbf{x}, y)\}$ where $\mathbf{x} = [1 \; x_1 \; \ldots \; x_d]^\top$. We define a loss for this dataset as

$$\mathcal{L}(\mathbf{w}, \mathcal{D}_{test}) = (y - h_{\mathbf{w}}(\mathbf{x}))^2 = |y - \hat{y}|^2$$

We define $h_{\mathbf{w}}(x) = \hat{y}$, and $|y - \hat{y}|$ is called a *residual*.
Now for a dataset containing n datapoints,

$$\textbf{Least Squares Loss}: \;\; \mathcal{L}(\mathbf{w}, \mathcal{D}_{train}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2 = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2$$

Note that the least squares loss is the mean of the squares of the residuals. We can also define a loss equal to the mean residual value.

$$\text{Mean Absolute Error Loss: } \;\; \mathcal{L}(\mathbf{w}, \mathcal{D}_{train}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

The Mean Absolute Error does not penalize high deviations as much as the mean squared loss.

Here we are interested in the least squared loss. We can vectorize the loss function as:

$$\boxed{\mathcal{L}(\mathbf{w}, \mathcal{D}_{\text{train}}) = \frac{\mathbf{1}}{\mathbf{n}}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_{\mathbf{2}}^{\mathbf{2}}}$$

where,

$$\mathbf{X} = \begin{bmatrix} \longleftarrow \mathbf{x}_1^\top \longrightarrow \\ \longleftarrow \mathbf{x}_2^\top \longrightarrow \\ \vdots \\ \longleftarrow \mathbf{x}_n^\top \longrightarrow \end{bmatrix}_{n \times (d+1)}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1}$$

4 Solving the optimization problem to find the best predictor

Minimizing the loss function to find our optimum hypothesis h^*, where

$$h^* = \arg\min_{h \in \mathcal{H}} \mathcal{L}(h, \mathcal{D}_{train})$$

[Note that the optimisation is performed only over the training dataset]
We may also define the objective in terms of the parameter $\mathbf{w}$ corresponding to h.

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathcal{L}(\mathbf{w}, \mathcal{D}_{train})$$

$$\mathbf{w}_{LS} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top x_i)^2$$

We set out to find a closed form solution for $\mathbf{w}_{LS}$ for d dimensional data: To find the optimum, we set the derivative[1] to zero. We may drop the $\frac{1}{n}$ term.

$$\begin{aligned}
&\nabla_{\mathbf{w}} \mathcal{L} = \frac{\partial \mathcal{L}(\mathbf{w}, \mathcal{D}_{train})}{\partial \mathbf{w}} = \left[\frac{\partial \mathcal{L}(\mathbf{w}, \mathcal{D}_{train})}{\partial w_i} \right]_{i=1}^{n} = \left[-2(y_i - \mathbf{w}^\top x_i) x_i \right]_{i=1}^{n} = 0 \\
&\implies 2(-\mathbf{X}^\top \mathbf{y} + \mathbf{X}^\top \mathbf{X} \mathbf{w}) = 0 \\
&\implies \mathbf{w}_{LS} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top Y
\end{aligned}$$

We can also derive this result with vector-derivative identities[1],

$$\begin{aligned}
&\nabla_{\mathbf{w}} \mathcal{L} = 0 \\
\implies &\frac{\partial}{\partial \mathbf{w}} \|\mathbf{y} - X\mathbf{w}\|_2^2 = \frac{\partial}{\partial \mathbf{w}} \left(\mathbf{y}^\top \mathbf{y} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{y} + \mathbf{w}^\top \mathbf{X}^\top X \mathbf{w} \right) = 0 \\
\implies &-2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X} \mathbf{w} = 0 \\
\implies &\mathbf{X}^\top \mathbf{X} \mathbf{w} = \mathbf{X}^\top \mathbf{y} \\
&\boxed{\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} (\mathbf{X}^\top \mathbf{y})}
\end{aligned}$$

Note that $\mathbf{X}^\top \mathbf{X}$ need not be invertible.

5 Homework

For 1D data, $h_{\mathbf{w}}(x) = w_0 + w_1 x$, show that:

$$\mathbf{w}_1{}^* = \frac{\Sigma (x_i - \overline{x})(y_i - \overline{y})}{\Sigma (x_i - \overline{x})^2}$$

where $\overline{x} = \frac{\Sigma x_i}{N}$ and $\overline{y} = \frac{\Sigma y_i}{N}$.

[1] There are two conventions for the derivative of a scalar by a vector, i.e., whether the result is a row or column. Here we will stick with the latter.