Data 100, Discussion 9

Suraj Rampure

Wednesday, October 23rd, 2019

Data 100, Discussion 9

Suraj Rampure

Wednesday, October 23rd, 2019

Agenda

- Multiple Regression
- Solution to OLS
- One-hot Encoding

Multiple Regression

Our model is

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$$

- $x_1, x_2, ..., x_p$ are called "features", "covariates", "explanatory variables", columns"
- $\beta_0, \beta_1, ..., \beta_p$ are called "weights"
- y is called the "response variable"

Our goal is to model the relationship between explanatory variables and our response variable. More concretely, our goal is to find values of $\beta_0, \beta_1, ..., \beta_p$ that minimize

$$R(\beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

We denote the optimal values by $\hat{\beta}_0$, $\hat{\beta}_1$, ..., $\hat{\beta}_p$, or more generally, by $\hat{\beta}$.

Features

Note: In order for us to perform linear regression, our model only needs to be **linear in terms of the weights**, not necessarily in terms of the features!

For example, consider the example from Lab 9. We want to predict `mpg` given `horsepower` (x_1), `model year` (x_2) and `acceleration` (x_3). The following are both linear:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 \sin x_3 + \beta_3 e^{x_1 x_2 x_3}$$

On the other hand, the following model is **not linear** – why?

$$y = \beta_0 + \beta_1 x_1^{\beta_2} + \beta_2^2 x_1 x_2 x_3$$

not linear (w.r.t. β)

not linear

Matrix Formulation

Our model is $y = X\beta$:

bias column

X : data matrix

β : weight vector

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{13} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & x_{23} & \cdots & x_{2p} \\ 1 & x_{31} & x_{32} & x_{33} & \cdots & x_{3p} \\ & & & & \vdots & \\ 1 & x_{n1} & x_{n2} & x_{n3} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix}$$

- n data points (observations). Each corresponds to one row.
- p features, plus a bias column. Each column is a "feature".
- Almost always, $n > p$

$$R(\beta) = \frac{1}{n} \|y - X\beta\|_2^2$$

Question – How do we find the optimal $\hat{\beta}$?

1. Through calculus – take the gradient, set it equal to 0.
2. Through geometry

Calculus Derivation

$$R(\beta) = \frac{1}{n}||y - X\beta||_2^2 = \frac{1}{n}\left((y - X\beta)^T(y - X\beta)\right)$$

$$= \frac{1}{n}\left(y^T y - y^T X\beta - (X\beta)^T y - \beta^T X^T X\beta\right)$$

$$= \frac{1}{n}\left(y^T y - 2(X^T y)^T \beta - (X\beta)^T(X\beta)\right)$$

Taking the gradient and setting it equal to 0:

$$\nabla R(\beta) = \frac{1}{n}\left(0 - 2X^T y - 2X^T X\beta\right) = 0$$

$$\Rightarrow X^T X\beta = X^T y$$

$$\Rightarrow \hat{\beta} = (X^T X)^{-1} X^T y$$

Motivating One-Hot Encoding

Our previous formulation assumes that all of our data was **numerical**. What if we want to somehow incorporate a **categorical** feature?

As an example, suppose we collect the heights, eye colors, and weights of several students, and we want to try and predict weight, given height and eye color.

$$X = \begin{bmatrix} 72 & \text{brown} \\ 65 & \text{black} \\ 67 & \text{blue} \\ \vdots & \vdots \\ 70 & \text{blue} \end{bmatrix} \qquad y = \begin{bmatrix} 150 \\ 98 \\ 102 \\ \vdots \\ 204 \end{bmatrix}$$

How can we fix this?

**For the purposes of this example, assume everyone's eye color is either brown, black, or blue.*

One-Hot Encoding

$$\begin{bmatrix} 72 & \text{brown} \\ 65 & \text{black} \\ 67 & \text{blue} \\ \vdots & \vdots \\ 70 & \text{blue} \end{bmatrix} \quad \Rightarrow \quad X = \begin{array}{c} \begin{matrix} & \text{brown} & \text{black} & \text{blue} \end{matrix} \\ \begin{bmatrix} 72 & 1 & 0 & 0 \\ 65 & 0 & 1 & 0 \\ 67 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 70 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Our model now looks something like

$$\hat{y} = \beta_1 \cdot \text{height} + \beta_2 \cdot (\text{color} = \text{brown}) + \beta_3 \cdot (\text{color} = \text{black}) + \beta_4 \cdot (\text{color} = \text{blue})$$

- In short, we create one column for each of our categories.
- For each row, exactly one of these columns contains the value 1, and the rest all contain the value 0.

One-Hot Encoding with an intercept term

Note: This is not relevant for this week's worksheet, but it was covered in Lecture 16, and is certainly relevant in the future.

When performing one-hot encoding, there's an additional consideration we need to make when including a bias column (i.e. a column of all 1s).

$$X = \begin{bmatrix} 72 & 1 & 0 & 0 & 1 \\ 65 & 0 & 1 & 0 & 1 \\ 67 & 0 & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \\ 70 & 0 & 0 & 1 & 1 \end{bmatrix}$$

What is the problem with this matrix?

not full rank!

One-Hot Encoding with an intercept term

$$X = \begin{bmatrix} 72 & 1 & 0 & 0 & 1 \\ 65 & 0 & 1 & 0 & 1 \\ 67 & 0 & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \\ 70 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Here, X **is not full rank!**

- This is because $\text{col } 2 + \text{col } 3 + \text{col } 4 = \text{col } 5$, i.e. one of our columns can be written as a linear combination of the others (this is true because, for each row, **exactly one** of $\{\text{col } 2, \text{col } 3, \text{col } 4\}$ is set to 1, and the rest are all 0)

Solution: Drop one of $\{\text{col } 2, \text{col } 3, \text{col } 4\}$.

- Why is this valid? Are we losing any information?

can recreate missing column: $1 - \text{col } 2 - \text{col } 3 = \text{col } 4$

→ same information, just presented in a way that's full rank