Linear Regression

Alan Liang

May 2018

1 The Idea

At its core, linear regression tells us a linear relationship between variables. This relationship allows us to conduct both prediction and causal inference.
Generally, any regression line will be of the form:

$$Y = \alpha + \beta X$$

Essentially, this depicts a linear relationship between the explanatory variable X and the outcome (dependent variable) Y. For every one unit of increase in X, we can associate it with a β unit increase in Y. The α on the other hand is the y-intercept, which essentially means the value of Y when there is no X, or $X = 0$.
If there are many variables $X_1, X_2 \cdots$ to account for that may contribute to the outcome variable:

$$\hat{Y} = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots$$

A quick note on notation: generally, we will use a hat symbol to denote a prediction/estimated value, in this case Y is being estimated by X. Therefore, the relationship between Y and $\hat{Y}$ is the difference of the error (residual) term:

$$Y = \hat{Y} + e = \alpha + \beta X + e$$

2 Doing Linear Regression

Generally, we will use the least squares method to conduct linear regression. I won't go into too much detail, but the idea is that we want to minimize the root mean square of the residuals:

$$\min \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{n}}$$

Since at any data point $\hat{y}_i = \alpha + \beta x$:

$$\min \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{n}} = \min \sqrt{\frac{\sum_{i=1}^{n}(y_i - (\alpha + \beta x))^2}{n}}$$

You could take the derivative of this, but it gets messy.
So instead, in data 8 we've abstracted that away to a `minimize` function, which when we pass in a function, will return us the coefficients that create the minimum value for that function (how does this work? gradient descent! Take EE 127 or DS 100).

2.1 Using `minimize`

In general, the function we build to pass into minimize will calculate and return the RMSE of the residuals.

```
def RMSE(alpha, beta):
        x_values = table.column("x") #column of x values to conduct prediction
        actual_y = table.column("actual_y") #column of actual y values
        predicted_y = alpha + beta * x_values #column of predicted y values
        rmse = np.mean((actual_y - predicted_y) ** 2) ** (1/2)
        return rmse
```

Keep in mind that here we are only doing column (array) calculations.

3 Multiple Regression and Controlling for Covariates

When many variables are involved in the linear regression, the idea of partial slopes comes in. Essentially, a partial slope for some variable X signifies that, assuming all other variables stay the same, a change in 1 unit of X will be associated with the slope's change in the outcome. If you've taken a multivariate calculus course, the intuition should be pretty straightforward.
We can leverage this definition and utilize multiple regression to control for other potential confounding variables. For example, say you want to know the effect of square footage X_1 on the price of a house Y. A possible confounding factor is the amount of bedrooms in the house X_2. With multiple regression, you can create an equation that takes both of these variables into account.

$$Y = \alpha + \beta X_1 + \gamma X_2$$

Since the partial slope β is defined to be the change on Y from X if no other variables (i.e. γ) change, we would have essentially controlled for confounding variable X_2: the treatment effect can be measured by β. In terms of our example, β would measure the increase in price for ever 1 sqft increase in area, assuming that the house has the same amount of bedrooms (X_2 is kept constant).
As you can probably see, regression provides a natural method to take into account of and hence control for covariates.

3.1 Regression, Matching, and Causal Inference

Conceptually, regression is similar to matching in that both utilize the selection on the observables assumption, which assumes that conditioning on the observable variables is as good as random assignment. Under this assumption, we are then freely able to determine the causal effect of the treatment variable X_1 on the outcome Y.
In general, regression is easier to implement than matching: instead of 'guessing' the treatment effect from comparing similar neighbors, we are predicting the treatment effect from a linear regression formed by the entire data set. Usually, both will yield have the same results.
However, there is a caveat with regression: linear regression, as the name implies, assumes that all variables have a linear relationship. This may not be true. Hence, we may have to transform the data (for example by using logarithms) to make it linear first.