LINEAR REGRESSION WITH MULTIPLE PREDICTORS

ETHAN LEVIEN

Contents

1. Multiple predictor linear regression

The real power of regression comes when we work with models of the form

$$Y = b + \sum_{i=1}^{K} a_i X_i + \epsilon \tag{1}$$

$$\epsilon \sim \text{Normal}(0, \sigma_\epsilon) \tag{2}$$

where X_i is a set of K predictor variables. If we want to think about this in terms of conditional averages, then

$$Y|(X_1 = x_1, \ldots, X_K = x_K) \sim \text{Normal}\left(b + \sum_{i=1}^{K} a_i X_i, \sigma_\epsilon\right) \tag{3}$$

This is the simplest generalization of the single-predictor regression model to work with multiple predictors, although as we will see it is not the only generalization. We now want to answer all the questions we asked for the original regression model in the context of this model, such as:

(1) What assumptions are we making and how do we interpret the parameters a_i?
(2) What are estimators of the parameters from data?
(3) How accurate is our model at predicting new Y values based on X values?

1.0.1. *Multiple predictors in python.* Let's start by seeing how to work with multiple predictors in python The first step is to get the predictor variables in the correct format for statsmodels. Statsmodels wants us to input a multidimensional arrray

$$X = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} \\ 1 & x_{2,1} & x_{2,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{n,1} & x_{n,2} \end{bmatrix} \tag{4}$$

The ith column contains the predictors that go with our ith observation y. This will tell statsmodels to also include a constant term (the intercept) β_0 in our regression.

The following code will get our data in this format:

```
> X = sm.add_constant(np.transpose(np.array([x_hs,x_iq])))
```

Example 1. *Our first regression with multiple predictors*

2. Interpretation and estimation of the parameters

In order to interpret the parameters, it's easiest to work with just two predictors:

$$Y = b + a_1 X_1 + a_2 X_2 + \epsilon. \tag{5}$$

Let start by just looking at the deterministic equation:

$$y = b + a_1 x_1 + a_2 x_2 \tag{6}$$

This describes a flat surface in two dimensions. If we make a slide through the surface in the x_1 direction and look it at from the side, we see a line with slope a_1 (and similarly for x_2). Now back to the regression model. We can understand a_1 is

Date: April 2022.

the slope of Y vs. X_1 for fixed (conditioned on) X_2. **The fact that it doesn't matter which value of X_2 we condition is an assumption of the model**. Mathematically, we can write

$$a_1 = \mathbb{E}[Y|X_1 = (x+1), X_2] - \mathbb{E}[Y|X_1 = x, X_2]. \tag{7}$$

It is important that we condition on BOTH variables?

You might guess the coefficient a_1 is also $\mathrm{cov}(Y, X_1)/\sigma^2_{x_1}$. After all, if we look a slice of the 2D planer function $y(x_1, x_2)$ along the x_1 direction, we get the same slope for all x_2. It stands to reason if we look at only the points in the x_1-y plane our regression slope would be a_1. **This argument assumes that when we change x_1, x_2 does not also change**. This is best understood with an example

Example 2. *Understanding the multiple predictors regression slopes*

The important thing is that when we increase x_1 we are ALSO increasing x_2.

If the usual relationship in terms of the covariance doesn't hold, is there a more general relationship expression for a_1 in terms of conditional averages. The answer is, of course, yes! To get there, we need to us some linear algebra which is beyond the scope of these notes. If you are interested, it goes something like this:

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sigma^2_{x_1} & \mathrm{cov}(X_1, X_2) \\ \mathrm{cov}(X_1, X_2) & \sigma^2_{x_2} \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{cov}(X_1, Y) \\ \mathrm{cov}(X_2, Y) \end{bmatrix} \tag{8}$$

$$= \frac{1}{\sigma^2_{x_2}\sigma^2_{x_1} - \mathrm{cov}(X_1, X_2)^2} \begin{bmatrix} \sigma^2_{x_2} & -\mathrm{cov}(X_1, X_2) \\ -\mathrm{cov}(X_1, X_2) & \sigma^2_{x_1} \end{bmatrix} \begin{bmatrix} \mathrm{cov}(X_1, Y) \\ \mathrm{cov}(X_2, Y) \end{bmatrix} \tag{9}$$

After using the formula for the inverse of 2×2 matrix, we obtain

$$a_1 = \frac{\mathrm{cov}(X_1, Y)\sigma^2_{x_2} - \mathrm{cov}(X_2, Y)\mathrm{cov}(X_1, X_2)}{\sigma^2_{x_2}\sigma^2_{x_1} - \mathrm{cov}(X_1, X_2)^2} \tag{10}$$

$$= \frac{\mathrm{cov}(X_1, Y) - \mathrm{cov}(X_2, Y)\mathrm{cov}(X_1, X_2)/\sigma^2_{x_2}}{\sigma^2_{x_1} - \mathrm{cov}(X_1, X_2)^2/\sigma^2_{x_2}} \tag{11}$$

This tells us how a_1 can be estimated from data! Notice that if all the variances are equal to one:

$$a_1 = \frac{1}{1 - \rho_{1,2}}(\rho_1 - \rho_{1,2}\rho_2) \tag{12}$$

where $\rho_{1,2}$ is the correlation coefficient between X_1 and X_2.

Exercise 1: *Test score data*

Exercise 2: *More on test scores*

This can all be generalized to the situation where we have many predictors. The general formula for the regression coefficient would be:

$$\begin{aligned} a_i = \mathbb{E}[Y|X_1, \ldots, X_{i-1}, X_i = x_i + 1, X_{i+1}, \ldots, X_K] \\ - \mathbb{E}[Y|X_1, \ldots, X_{i-1}, X_i = x_i, X_{i+1}, \ldots, X_K] \end{aligned} \tag{13}$$

2.0.1. *The sample distribution of coefficients.* Just as before, we want to understand what the sample distribution of the coefficients looks like. In the multiple predictor case, this becomes more complicated because we have multiple parameters.

Exercise 3: *Understanding multivariate sample distribution*

3. Dealing with categorical data

One situation in which models with multiple predictors frequently arrises is when trying to predict a Y variable based on categorical predictors, such as race. In this case, we need to transform the categories into numerical values. For example, if there are two cataogies, we map our variable to 0 or 1. If we have 3 categories, we might first think to map them to 0, 1 and 2. This has a problem though: A chance from 1 to 2 should not necessirility correspond to a change from 0 to 1. **There is no ordering of the x values**. Thus, instead we introduce a new variable, which is 1 if our data point is in the third category and zero otherwise. Do you see what the problem would be if we have 3 X variables, one for each category?

In order to take a categorical variable and transform it into a set of indicator variables in python, we use

```
> get_dummies
```

Example 3. *Working with categorical data*