Unit 5: Nonlinear models and overparamaterized linear regression

Ethan Levien

November 10, 2025

Contents

5.1 Interactions terms in linear regression models

5.1.1 Interactions as new predictors

The important assumption of the multiple predictor regression models we have seen so far is that the "effect" of one predictor does not depend on the value of the other. Here are some examples where this could be violated:

- The difference in test scores between kids whose mothers did and and did not go to high school depends on their mother's score on the iq test.
- The association between earnings and height depends on gender, e.g. being taller tends to give men a larger advantage than women
- The effect of a drug for treating covid depends on whether someone has had covid before.

We call the dependencies described above interactions. It turns out it is possible to include interactions within the regression modeling framework we have already introduced, as is illustrated by the following example. Let's work in the case of two predictors. If you'd like, you can refer to Example 2 to make this math more concrete and always replace X_1 with X_{hs} and X_2 with X_{iq}. In each of the cases above, what is going on is that the slope of $\mathbb{E}[Y|X_1, X_2]$ vs. X_1 for fixed X_2 is not a constant, β_1, but rather something that depends on X_2. The simplest way to account for this is to let effect of X_2 on the slope be linear in X_2, so instead of β_1 being the slope of $\mathbb{E}[Y|X_1, X_2]$ vs. X_1 we would have $\beta_1 + J_{1,2}X_2$ be this slope, but this leads to a regression model with a nonlinear term:

$$Y = \beta_0 + (\beta_1 + J_{1,2}X_2)X_1 + \beta_2 X_2 = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + J_{1,2}X_1X_2 + \epsilon$$

We call X_1X_2 an interaction term, but this isn't linear anymore and the regression coefficients are also don't have the same interpretation. How do we deal with this? The idea is that we can create a new predictor from our already existing predictors which accounts for the interactions. To this end, consider the two predictor regression model written now as

$$Y = \beta_0' + \beta_1' X_1 + \beta_2' X_2 + \epsilon.$$

I'm again using the $'$ to distinguish this model from an expanded model with a new predictor. The new predictor which accounts for the interaction will be $X_3 = X_1 X_2$! Note that X_3 is going to be correlated with both X_1 and X_2, but crucially it will not be perfectly correlated unless X_1 and X_2 are with each other. The new mode is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + J_{1,2} X_3 + \epsilon$$

Or written as a conditional distribution

$$Y|X \sim \text{Normal}(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + J_{1,2} X_3, \sigma^2).$$

Now how do we interpret the regression coefficients in the new model? If we write out the usual formula for the regression coefficient with no interactions, we get

$$\begin{aligned}
&\mathbb{E}[Y|X_1 = x+1, X_2] - \mathbb{E}[Y|X_1 = x, X_2] \\
&= (\beta_1(x+1) + \beta_2 X_2 + \beta_3(x+1)X_2) - (\beta_1 x + \beta_2 X_2 + J_{1,2} x X_2) \\
&= \beta_1 + J_{1,2} X_2
\end{aligned}$$

Here I've replaced X_3 with $X_1 X_2$. I words, $J_{1,2}$ is the additional difference in the regression slope between Y and X_1 with all other predictors fixed when X_2 is changed by one unit. If $J_{1,2}$ is positive (resp. negative) then increasing X_2 has a tendency to increase (resp. decrease) the difference between the conditional averages of Y, thus, the additional predictor $X_3 = X_1 X_2$ allows us to capture an interaction wherein the association between X_1 and Y depends on X_2. We can do the same calculation with X_1 fixed and X_2 changed to obtain

$$\mathbb{E}[Y|X_1 = x+1, X_2] - \mathbb{E}[Y|X_1 = x, X_2] = \beta_2 + J_{1,2} X_2$$

Now as always, let's make sure we understand how this translates into code and also I think the visualizations in this example are helpful.

Example 1 (Visualizing interactions)**.** Consider the regression model with interactions term $X_3 = X_1 X_2$ defined above, assuming that

$$\begin{aligned}
X_1 &\sim \text{Normal}(0, 1) \\
X_2 &\sim \text{Bernoulli}(1/2)
\end{aligned}$$

Assume $\beta_1 = 1.2, \beta_2 = 0$ and $\beta_3 = -1$.

Question: What are the slopes of $E[Y|X_1, X_2 = 0]$ vs. X_1 and $E[Y|X_1, X_2 = 1]$ vs. X_1? Plot these in colab.

Example 2 (Interaction in test score model)**.** Here we once again consider the test score data with high school education and IQ as predictors. In particular, we will fit the model

$$Y = \beta_0 + \beta_{\text{hs}} X_{\text{hs}} + \beta_{\text{iq}} X_{\text{iq}} + J_{\text{iq,hs}} X_{\text{iq}} X_{\text{hs}}$$

Question:

(a) What are the values and interpretations of the regression coefficients in the new model?

(b) What do the results tell us about the "effect" of IQ on test scores and how it is related to high school education?

Solutions: The output of `statsmodels` is

```
> ==================================================================================
>                  coef    std err          t      P>|t|      [0.025      0.975]
> ----------------------------------------------------------------------------------
> const        -11.4820     13.758     -0.835      0.404     -38.523      15.559
> mom_hs        51.2682     15.338      3.343      0.001      21.122      81.414
> mom_iq         0.9689      0.148      6.531      0.000       0.677       1.260
> hs_iq         -0.4843      0.162     -2.985      0.003      -0.803      -0.165
```

(a) The values and interpretations are as follows

- $\beta_0 \approx -11$ does not have a clear interpretation since it doesn't make sense to have 0 IQ.
- $\beta_{\text{iq}} \approx 1$ is the expected signed difference in scores between students whose IQ differs by 1 point and whose mothers did not go to high school.
- $\beta_{\text{hs}} \approx 51$ is the expected difference between scores of students whose mothers did and did not go to high school when IQ is zero – **much like β_0, this doesn't really make sense, so we don't have a clear interpretation of β_{hs}**
- $J_{\text{hs,iq}} \approx -0.5$ is the expected increase between the group of students whose mothers attended high school and those that did not in the difference between test scores of students whose mothers iq differs by one point. In other words, it is the increase in regression slope of scores vs. iq between the two high school groups. It is easier to interpret $\beta_{\text{iq}} + J_{\text{hs,iq}}$ which is the expected signed difference in scores between students whose IQ differs by 1 point and whose mothers did attend high school (this is the same as the interpretation of β_{iq} for student's whose mothers attended high school.
- It appears that the association between IQ and test scores is weaker among students whose mother's attended high school. This makes intuitive sense. If someone attended high school the subtle differences in cognitive ability that IQ is supposed to measure are not as relevant.

In the example above we saw that adding an interaction term can make the interpretation of the regression coefficients a bit clunky, even those regression coefficients that are not involved in an interaction. In particular, we ran into the problem that the interpretation of β_{iq} was lost. To remedy this, we can define the centered predictor

$$\Delta_{\text{iq}} = X_{\text{iq}} - \bar{X}_{\text{iq}}$$

and now we fit the model

$$Y = \beta_0 + \beta_{\text{hs}} X_{\text{hs}} + \beta_{\text{iq}} \Delta_{\text{iq}} + J_{\text{hs,iq}} X_{\text{hs}} \Delta_{\text{iq}} + \epsilon$$

Now β_1 has the interpretation as the average difference in scores among students whose mothers have the average iq. You can also standardize the predictor to get around the awkward interpretation of the regression coefficients.

5.1.2 Residual plots

We address the question of how we might identify when it is appropriate to add an interaction term to a model. In this case of two predictors we can easily visualize the data by making plots of the Y vs. X_1 slopes for different X_2 values (especially when one predictor is binary). With many predictors, it becomes less clear what plot might reveal some hidden interaction terms. We could of course go on adding every possible interaction term, but with many predictors this becomes in practical and leads to overfitting (as we will soon discuss). The basic idea of residual plots is that by plotting the difference between the observed y values and the prediction of the $E[Y|X]$, or

$$\epsilon_j = Y_j - \hat{\beta}_0 - \sum_{k}^{K} \hat{\beta}_k X_{k,j}$$

where $X_{k,j}$ is the value of predictor i at the jth data point. Our goal is to plot ϵ_j in a such a way that disagreement between our model and the data is revealed by this plot. In the instance of a single-predictor, we can simply plot ϵ_j as a function of the predictor X. If we notice that the residuals do not appear to follow a normal distribution, or that the variance and mean change, then we should be skeptical. The question is: **When we have multiple predictors, what do we plot on the horizontal axis?** The answer is to plot ϵ_j as a function of the predictors value of $E[Y|X]$; that is ϵ_j vs. $\hat{y}_j = \sum_i^K \hat{\beta}_i X_{i,j}$. The following examples is supposed to help us understand why.

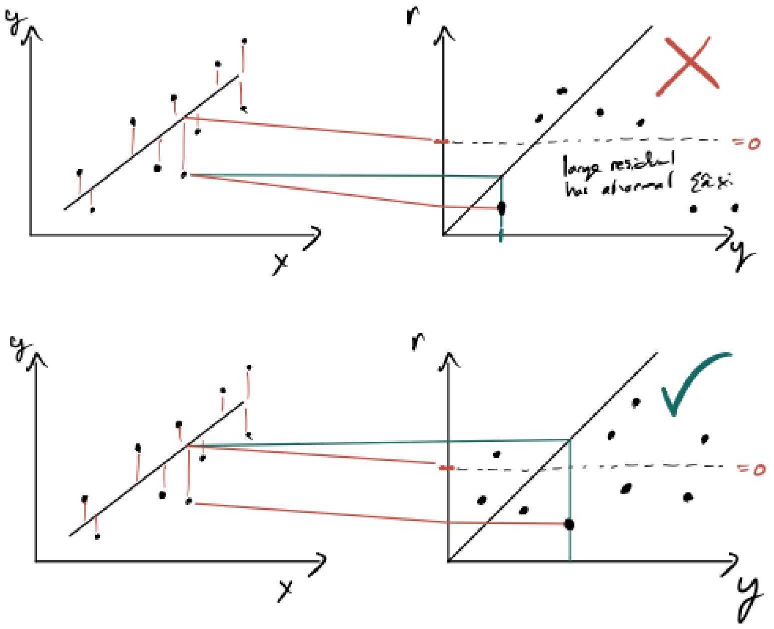

Figure 1: Correct and incorrect way to plot residuals against response variable

Example 3 (Residual plots)**.** Consider the regression model with two predictors and suppose the true parameter values are $\beta_0 = 0$,$\beta_1 = 0.2$,$\beta_2 = 20$ and $\sigma = 1$.

Question: Compare two ways of plotting the residuals

(a) Plot ϵ_i as a function of $\sum_k^K \hat{\beta}_k X_{k,i}$.

(b) Plot ϵ_i as a function of Y_i. Why does there appear to be a bias towards high values of ϵ_i for large Y_i that is not present in the first plot?

Solution: See colab notebook

To better understand what the residual plot tells us, recall

$$\epsilon_j \approx Y_j - \underbrace{E[Y|(X_1,\dots,X_K) = (X_{1,j},\dots,X_{K,j})]}_{\approx \hat{\beta}_0 + \sum_k^K \hat{\beta}_k X_{k,j}} \tag{1}$$

where K is the number of predictors. Thus, the distribution of ϵ_j is approximately

$$\epsilon_j \sim \text{Normal}(0,\sigma^2).$$

This tells us how the points should be distributed in the vertical direction. It **does not** say anything about the distribution of points in the horizontal direction, which is determined by the distribution of the predictors. Therefore, we expect a plot which is symmetric around the line ϵ_j for all values of $\hat{y}_j$ (our predicted values of Y), but any distribution in the horizontal direction is okay.

Now compare this to what would happen if we plotted Y_j on the horizontal axis, not $\hat{y}_j$. In this case, based on Equation 1 ϵ_j and Y_j are correlated. This is why we see a bias of the residuals for small/large Y_j in Example 3.

5.2 Other feature maps

Here, we discuss how to build more complex models and directly access their predictive power on out-of-sample data. In the context of interactions, we already saw how a model can be extended by defining a new predictor $X_3 = X_1X_2$. The more general idea that we can define a new predictor which is a function of the other predictors allows us to develop very complex and flexible models which nonetheless can be analyzed within linear regression framework. Here, we will formalize this, beginning with the case of a single predictor.

In general, the linear regression framework allows us to fit models of the form consider the model

$$Y|X \sim \text{Normal}(f(X),\sigma^2) \tag{2}$$

provided we can express $f(X)$ as a linear combinations of nonlinear functions of X. What I mean by this is that we can find function $\phi_1(x),\dots,\phi_K(x)$ such that

$$f(X) = \sum_{i=1}^{K} \beta_i \phi_i(X)$$

The functions $\phi_i(X)$ are often referred to as basis functions, or features in machine learning lingo. We can think of each $\phi_i(X)$ as a new predictor.

Example 4 (Simulating a nonlinear model)**.** Consider the conditional Gaussian model given in Equation 2 with

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2$$

This is simply a linear model once we define the new predictors

$$X_1 = \phi_1(X) = X$$
$$X_2 = \phi_2(X) = X^2.$$

Question:

(a) Generate data from this model

(b) Fit the data to the model using `statsmodels`.

Solution: See colab notebook

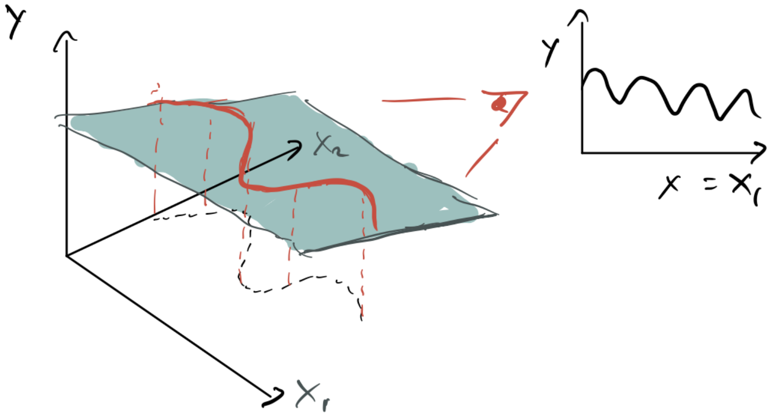

Figure 2: An illustration of how a nonlinear dependence on our predictor can be incorporated into the linear modeling framework by adding a feature.

Which functions ϕ should we use? The answer of course depends on the problem at hand. For example, we might know something about the physics of the data we are modeling. In some cases, we may select the ϕ so that the parameters a_i have clear interpretations (as is the case in linear regression model). The following illustrates such an example.

Example 5 (Mauna Kea Data)**.** Solution: See colab notebook

Next we will understand that within the linear regression framework we can fit very complicated nonlinear data. However, we need to be careful when adding new features to the model. The following example will illustrate how it is possible to "overshoot" and make our too complex. In the next set of notes we will dive much deeper into this, bridging the gap between machine learning and statistics.

5.3 Data partitioning and bias variance tradeoff

We saw from the previous example that R^2 is not very helpful if we want to decide how complex we'd like to make our model, since it is possible to explain all the variation in the Y values with a model which is too complex. In order to address this, we take the approach of cross validation. The basic idea of cross validation is to break our data up into two subsets: training set, which we use to fit the model, and a testing set, which we compare to the predictions of the fitted model.

Let's introduce some new notation. I will use D to refer to our entire data set:

$$D = (Y, X) = \{(X_1, Y_1), \dots, (X_N, Y_N)\}$$

$D^{\text{train}} = (Y^{\text{train}}, X^{\text{train}})$ and $D^{\text{test}} = (Y^{\text{test}}, X^{\text{test}})$ will represent the subsets of the data to be used to training (that is, fitting) the model and testing the fit respectively. We will assume that

$$D = D^{\text{train}} \cup D^{\text{test}}$$

Let N_{train} and N_{test} denote the number of points in each group. Let $\hat{y}(x, D)$ the prediction of $E[Y|X = x]$ using the fitted coefficients based on a data set D; that is

$$\hat{y}(x, D) = \sum_{i=1}^{K} \hat{\beta}_i \phi_i(x)$$

where $\hat{\beta}$ are the (usually least squares) fitted coefficients using the data in D.

Now let us define the training error

$$\hat{\sigma}^2_{\text{train}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} (\hat{y}(X_i^{\text{train}}, D^{\text{train}}) - Y_i^{\text{train}})^2$$

where the average is taken over different replicates of our data and $\hat{Y}_i$ is our prediction of $E[Y|X]$. Not that the training error tells us about how well our model does at predicting the same data we used to fit it, and therefore not surpisingly, it is closely related to R^2:

$$R^2 \approx 1 - \frac{\hat{\sigma}^2_{\text{train}}}{\sigma^2_Y}.$$

In particular, if we replace σ_Y with its approximation we get the R^2 for the training data. With enough training data this should just be R^2 for the model since $\hat{\sigma}^2_{\text{train}} \approx \sigma^2_\epsilon$.

In order to see how well our model does at predicting the points we did NOT use to fit it, we introduce the test error:

$$\hat{\sigma}^2_{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} (\hat{y}(X_i^{\text{test}}, D^{\text{train}}) - Y_i^{\text{test}})^2$$

Note that the fitted coefficients used to compute $\hat{y}_i(X^{\text{test}})$ come from fitting the model to the training data, even though we are evaluating $\hat{y}$ at the test points.

Example 6 (Cross validation on polynomial model)**.** Consider example 6 from the previous weeks notes.

Question: Plot the training and test error as a function of the number of parameters.

Solution: See colab notebook.

5.4 Overfitting

In order to understand U-shaped curve seen in Example 6, we first need to introduce some more general terminology and notation for talking about estimators (of which $\hat{y}$ is an example). Recall from Unit 3 that the mean-squared error of an estimator $\hat{\theta}$ of some quantity θ is

$$\text{MSE}_{\hat{\theta}} = E\left[(\hat{\theta} - \theta)^2\right] \tag{3}$$

Remember, θ could be a parameter, or it could be a value of a function, such as $f(x)$, that we would like to predict. For now, let's just think of $\hat{\theta}$ as any estimator. Also, remember that we proved the bias variance decomposition:

$$\text{MSE}_{\hat{\theta}} = \text{var}(\hat{\theta}) + E\left[\hat{\theta} - \theta\right]^2 \tag{4}$$

Now let's return to working in the context of a model of the form

$$Y|X \sim \text{Normal}(f(X), \sigma^2)$$

where $f(X)$ can be written as a linear combination of features

$$f(X) = \sum_{i=1}^{N} \beta_i \phi_i(X).$$

We will study the MSE in our predictions of the $f(x)$ given a dataset D, and hence we set $\hat{\theta} = \hat{y}(x, D)$ so that Equation 3 becomes

$$\text{MSE}_{\hat{y}(x,D)} = E_D\left[(\hat{y}(x,D) - f(x))^2 | x\right]$$

where the notation E_D is to remind us that the expectation is taken over the distribution of our data set D, while the predictor value is being conditioned on (I slightly abused notation and wrote x instead of $X = x$). The bias variance tradeoff is related to the test error $\hat{\sigma}^2_{\text{test}}$; however there are two differences between what the MSE is measuring and the test error: (1) In the test error the average is over different X points and (2) the test error measures the differences between the observed points not the true function. If N^{test} is sufficiently large, then

$$\hat{\sigma}^2_{\text{test}} \approx E_X[(\hat{y}(x,D) - Y)^2 | D], \quad Y = f(x) + \epsilon. \tag{5}$$

Hence the test error is the MSE for a fixed data set with the average being taken over D. For this reason we use E_X to remind us that the expectation with respect to the distribution of our predictor, and it is assumed that X is sampled independently from our data D. Note that I use D on the right hand side, when really the test error is obtained from D^{train} only, but since the total data D and the test error have the same distribution, we can simply write D and that makes the notation easier.

It can be shown that (see exercise)

$$E_X[(\hat{y}(x,D) - Y)^2 | D] = E_X[(\hat{y}(x,D) - f(x))^2 | D] + \sigma_\epsilon^2 \tag{6}$$

The second term σ_ϵ^2 does not depend on x or D, or the model we are using to fit the data. It only depends on the true data distribution. We can then relate the first term

$$E_D[E_X[(\hat{y}(x,D) - f(x))^2 | D]] = E_X[\text{MSE}_{\hat{y}(X,D)}] \tag{7}$$

and therefore

$$E_D[\hat{\sigma}^2_{\text{test}}] = E_X[\text{MSE}_{\hat{y}(X,D)}] + \sigma_\epsilon^2. \tag{8}$$

In practice, we only observe on a single dataset, but we can, but resampling the training data, effectively average over D to obtain an approximation of $E_D[\hat{\sigma}^2_{\text{test}}]$. Due to the bias variance decomposition,

$$E_D[\hat{\sigma}^2_{\text{test}}] \approx E_X[\text{var}(\hat{y}(X,D))] + E_X[(E_D[\hat{y}(X,D)] - f(X))^2] + \sigma_\epsilon^2. \tag{9}$$

In the next example we see what this looks like in practice. The point here is that up to a constant shift by σ_ϵ^2 will on average obey the same bias variance decomposition as we are used to from the MSE of any estimator. The variance is bias terms have more complex interpretation because they involve both averages over the data distribution and the predictor, but nonetheless reveals the same underlying idea. This is illustrated in the next example.

Example 7 (Plotting Bias and variance)**.** Let's continue with Example 6 and try to illustrate the bias variance tradeoff by computing these (or really approximations to them) separately.

Solution: See colab notebook.

We now summarize some observations we've made thus far:

- In a regression model, as we add more coefficients, eventually the model becomes more and more flexible, in the sense that is can describe more different types of data sets. Here, by "describe", we mean that it can be fit to those data sets with a high value of R^2 or low training error. With enough features, a model can perfectly interpolate between the data points, meaning $\hat{y}(X_i, D) = Y_i$ for each data point (X_i, Y_i) when the model is fit on all our data points.

- Unlike the training error, the test error, $\hat{\sigma}^2_{\text{test}}$ does not simply increase as we make the model more complex, rather it has a U-shape. Thus, there is an optimal model size at which our model's predictions of new data points – that is, data points outside the set we used to fit it – is best.
- The *U*-shape can be understood in terms of bias and variance. We know this because the mean-squared error, which is the "math world" version of $\hat{\sigma}^2_{\text{test}}$, can be decomposed into a bias and variance term. The variance term tells us how variable the predictions of our model will be when we fit it to different training sets, while the bias tells us how much its predictions. will differ, on average, from real data.

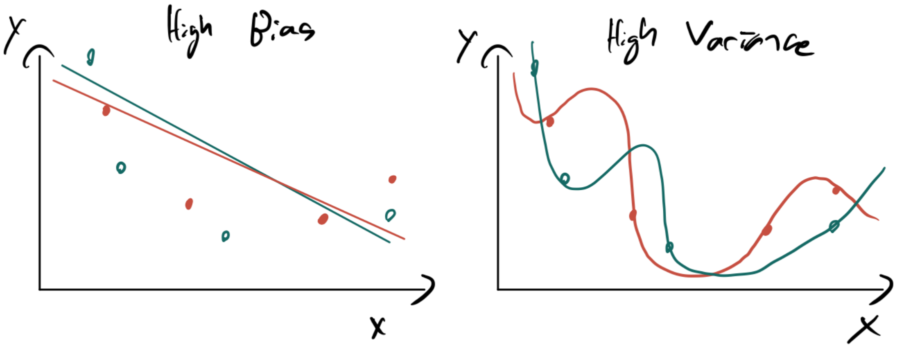

Figure 3: Bias and variance illustrated with fits to two different datasets (blue and red) drawn the from the same distribution

5.5 Orthogonality and Fourier analysis

Sometimes we have a particularly hypothesis about what features my contribute to an signal we are analyzing – for example, the CO2 data from last week we suspected there was a linear trend and yearly trend superimpose together. This was reasonable because we know that weather tends to follow a yearly cycle. But in general, if we don't have such knowledge how should we select the functions ϕ? It is often to our advantage to select basis function ϕ_i so that our predictors are not correlated. That is, if $\phi_i(X)$ and $\phi_j(X)$ are thought of as random variables (the randomness comes from X) then we want them to be uncorrelated.

Throughout our discussion, we will assume that $E[\phi_i(X)] = 0$. We can easily make this so by subtracting the means of our features. Therefore, we would like to find ϕ_i such that

$$\text{cov}(\phi_i(X)\phi_j(X)) = 0 \quad i \neq j. \tag{10}$$

We this holds for some distribution of X, we say that ϕ_i and ϕ_j are orthogonal with respect to this distribution. Thus, by orthogonal we just mean uncorrelated.

Example 8 (Sin series)**.** Let us now work with the particular example where

$$X \sim \text{Uniform}(-L, L). \tag{11}$$

Even if our X points are not random, but are say evenly spaced on the interval $[-L, L]$, a random sample from the uniform distribution will be statistical similar to some randomly selected evenly spaced X points. Thus, we can think of the uniform distribution as approximating the spread of our X data with a random variable.

Now consider the basis functions

$$\phi_j(x) = \sin\left(\frac{\pi x j}{L}\right).$$

Question: Show using simulations that ϕ_j are orthogonal with respect to the uniform distribution on $[-L,L]$ (Equation 11).

Solution: See colab notebook.

Notice that if ϕ_j are orthogonal, then we can express the regression coefficients as

$$\beta_j = \frac{\text{cov}(Y,\phi_j(X))}{\text{var}(\phi_j(X))} = \frac{E[Y\phi_j(X)]}{E[\phi_j(X)^2]}$$

and hence our fitted coefficients from N data points can be approximated as

$$\hat{\beta}_j \approx \frac{\sum_{i=1}^N \phi_j(X_i)Y_i}{\sum_{i=1}^N \phi_j(X_i)^2} \tag{12}$$

This suggests $\hat{\beta}_j$ should depend only weakly how many of the features ϕ we have included in our model! This is consequence of orthogonality. In contrast, in the earlier example where we used polynomial features adding a new predictor, say X^4, would dramatically change the fitted value of β_2. However, Now let's return to Example 8. A limitation of this model is that it only permits us to model "odd functions" that is, functions for which $f(x) = -f(x)$, as each ϕ_j has this property. It is therefore desirable to add addition features which permit this, but we'd like them to still be orthogonal. To this end, we introduce a new set of features to the model given by the cosine functions:

$$\cos\left(\frac{\pi x j}{L}\right).$$

Our new model, called a Fourier series, is given by

$$f(x) = \beta_0 + \sum_{i=1}^{K} \beta_i \sin\left(\frac{\pi i x}{L}\right) + \alpha_i \cos\left(\frac{\pi i x}{L}\right)$$

where we are using α_i to represent the coefficients of the cosine terms. Fourier series are on of the most important models in data science and engineering. It can be proved that as $K \to \infty$ we can approximate essentially ANY function with a series of this form.

Example 9 (Working with Fourier series in python)**.** Suppose we have data from the model with

$$f(x) = \sin(3\pi x) + \sin(10\pi x)$$

where $X_i = (i-1)/N$ (meaning the X points are evenly spaced in $[0,1]$.

Question: For different values of σ^2, generate data from this model and fit it to a Fourier series with $K = 100$ terms.

Solution: See colab notebook.

The process of computing the coefficients $\hat{\beta}_j$ and $\hat{\alpha}_j$ for the Fourier series is called a Discrete Fourier Transform. Usually DFT refers to the cases where the X_i are equally spaced. In this case, the orthogonality condition (Equation 10) is true is "data world", not just "math word" – what I mean by this is that for equally spaced data points.

To be precise, consider the predictor data on the interval $[0,L]$ given by

$$X_i = \frac{L(i-1)}{N-1}$$

for $i = 1, \dots, N$. Thus $X_1 = 0$, $X_2 = L/N$, $X_3 = 2L/N$ and $X_N = L$. We could generate these points with `np.linspace`. The following theorem tells us that the empirical, or sample covariance between the sin features, is exactly zero for these predictors.

Theorem 1.

$$\sum_{i=1}^{N} \sin\left(\frac{2\pi j X_i}{L}\right) \sin\left(\frac{2\pi k X_i}{L}\right) = 0 \quad k \neq j$$

Often we want to summarize how different frequencies are represented in our data, but we don't particularly care about whether they come form the sin or cos terms. To achieve this, one uses the power spectrum density, also known as the peridogram,

$$P_j = \beta_j^2 + \alpha_j^2.$$

The power spectrum density is a fundamental object in signal processing, and it essentially tells us how "wobbly" a signal is.

Example 10 (Periodogram)**.** Question: Compute the periodogram of the data generated in Example **??** and confirm the same periodogram can be generated with the `periodogram` function from the `scipy.signal` library. This is neat and not often made point, that this fundamental structure from signal processing is in fact coming from fitting a linear regression model with least square!!

Solution: See colab notebook.

Exercises

Exercise 1 (Car brands and mpg)**:** In this exercise we will consider the data set containing information about cars and their miles per gallon. This can by loaded by

```
> data = pd.read_csv("https://raw.githubusercontent.com/intro-stat-learning
>                       /ISLP/main/ISLP/data/Auto.csv",encoding = "ISO-8859-1")
> data["name"] = [name.split()[0] for name in data["name"].values]
```

The second line takes the original names (which are the specific models – e.g. Toyota Yaris) and extracts only the brand name (e.g Toyota). We are going to study which brands have the best mpg. Some brands tend to make larger and heavier cars (e.g. pickup tricks) which will have worse mpg, but we want to understand how brands compare within a certain type of car. To determine this we need to control for other factors, such as the the year and weight.

(a) Using all the columns **except** `origin` and `displacement` (since it's not obvious what the units are), write down the regression model which you want to fit to this data to address the question posed in the problem instruction. Assume there are no interactions. Provide an interpretation of each regression coefficient.

(b) Fit the regression model to the data.

(c) What are the 5 best brands for mpg within the same type of car (weight, horsepower etc.).

(d) Now try to improve the model by adding interaction terms. Do you find any interactions which are statistically significant?

Exercise 2 (Marginal regression in interactions model)**:** Consider the probability model

$$
\begin{aligned}
X_1 &\sim \text{Normal}(0, \sigma_1^2)\\
X_2 &\sim \text{Normal}(0, \sigma_2^2)\\
Y|(X_1, X_2) &\sim \text{Normal}(\beta_1 X_1 + \beta_2 X_2 + J_{1,2} X_1 X_2, \sigma^2)
\end{aligned}
$$

(a) Derive the distributions of $Y|X_1$ and $Y|X_2$. Hint: These conditional distributions are both normal, so you only need to determine the mean and variance to find the distributions.

(b) When does the probability model stated in the problem define regression models for Y vs. X_i, $i = 1, 2$? That is, if we ignore one of the predictor variables do obtain a single predictor linear regression model for the other?Would this be true if the predictors did not have zero mean?

Exercise 3 (Predicting the residual plot based on interaction model)**:** Suppose we have 200 data points generated from the following model

$$Y = 4X_1 - 2X_2 + 4X_1X_2 + \epsilon \tag{13}$$

where $\sigma = 0.2$, x_1 is continuous predictor which is uniformly distributed between -1 and 1 and X_2 is a binary predictor (e.g. a Bernoulli random variable). You can assume $X_2 = 0$ for about half the data points. The goal of this problem is to build your intuition about residual plots.

(a) **Without actually fitting a regression**, describe in detail what the residual plot would look like if we fit this data to a linear regression model with NO interaction term. To do so, follow the following procedure

- First, think about what the data looks like when $X_2 = 0$ and $X_2 = 1$ separately. In each case, sketch the regression line and make note of how much variation there is around these lines to get an idea of what the cloud of (X_i, Y_i) points will look like.
- Now consider what the fitted regression line will be based on this picture. What is a very rough estimate of the slopes $\hat{\beta}_1$ and $\hat{\beta}_2$?

- To get a sense for what the residuals look like, take the difference between the true model and this line.

(b) Confirm you answer with simulations.

Exercise 4 (Drug interactions)**:** When treating microbial infections and cancer, combinations of drugs can perform better than individual drugs. However, it can be difficult to identify which combinations are optimal for the reason that identifying very "high order" interactions is difficult. In order to understand the best way to combine M drugs, we construct a regression where Y is the "effect" of the drug and X_i is a Bernoulli random variable representing whether or not the ith drug is present or not. We want to consider the possibility

$$Y = \sum_{i=1}^{M} \beta_i X_i + \sum_{i=1}^{M}\left(\sum_{j>i}^{M} J_{i,j} X_i X_j\right) + \sum_{i=1}^{M}\left(\sum_{j>i}^{M}\sum_{k>j}^{M} J_{i,j,k} X_i X_j X_k\right) + J_{1,2,\dots,M} X_1 X_2 \cdots X_M + \epsilon$$

For example, with $M = 3$, we would have

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + J_{1,2} X_1 X_2 + J_{1,3} X_1 X_3 + J_{2,3} X_2 X_3 + J_{1,2,3} X_1 X_2 X_3 + \epsilon$$

(a) Suppose $M = 3$ and

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ J_{1,2} \\ J_{1,3} \\ J_{2,3} \\ J_{1,2,3} \end{bmatrix} = \begin{bmatrix} 1.2 \\ -0.8 \\ -0.11 \\ 3.48 \\ -2.62 \\ 1.03 \\ 1.66 \end{bmatrix}$$

What is the interpretation of each coefficient?

(b) What is the optimal treatment, meaning which combination of drugs 1, 2 and 3 should we use to maximize Y? There are different ways you can approach this. One way is to make a list of each (X_1, X_2, X_3), compute Y for each one and the find the index of the maximum Y value (using a for loop or `argmax`).

(c) Now additional suppose that $\sigma^2 = 1$. By generating simulated Y values with these parameters for different values of N, determine how many data points are needed to reliably find that all interactions coefficients have p-values below 0.05

(d) Perform the same experiment as in (c) but fit the data to a model with no interactions. What do you find? How does adding the interaction terms influence the p-values.

Exercise 5: Consider two binary predictors $X_1, X_2 \in \{0, 1\}$ with joint distribution given by the table of probabilities:

$P(X_1, X_2)$	$X_2 = 0$	$X_2 = 1$
$X_1 = 0$	0.2	0.3
$X_1 = 1$	0.1	0.4

(a) Are X_1 and X_2 independent?

(b) If a interaction term is added be creating a predictor $X_3 = X_1 X_2$, then find the correlation coefficient between X_3 and X_1 and X_3 and X_2?

(c) Now generate data from a model (with the interaction term) and fit it to a regression model (also with the interaction term). Looking at the correlation of $\hat{\beta}_1$ with $\hat{J}_{1,2}$, is it consistent with your calculation in part (b)?

Exercise 6 (Mean-squared error)**:** Prove formula Eq. 6.

Exercise 7 (Laplace's rule)**:** Consider a Bernoulli random variable

$$X \sim \text{Bernoulli}(q).$$

(You can assume this means $P(X = 1) = q$). If we have samples $X_1, \dots, X_N$ for X then we have seen that a consistent and unbiased estimator of q is

$$\hat{q} = \frac{Y}{N}, \text{ where } Y = \sum_{i=1}^{N} X_i.$$

An alternative estimator, called Laplace's rule of succession, is

$$\hat{q}_L = \frac{Y+1}{N+2}.$$

The motivation of for defining $\hat{q}_L$ is as follows: Think of X as a biased coin. If we know that it is possible for to roll a heads or a tails, then we should include this information in our estimator. However, using the original estimator, if we roll a sequence of only heads (or tails), we will estimate that $q = 1$ (or $q = 0$). To correct for this, we pretend we have two additional observations (hence the $N + 2$ in the denominator) and that one is heads and one is tails (hence $Y + 1$ in the numerator). This is a simple example where we are incorporating *prior* information into our estimator – that is, information beyond what is present in the data and our model. In this case, that prior information is that the coin has two sides and could land on either one, however unlikely that might be.

(a) Derive formula for the mean-squared error,

$$\text{MSE}_{\hat{q}_L} = E\left[(\hat{q} - q)^2\right]$$

and decompose it into the variance and the squared bias. Your formulas should be in terms of q and N.

(b) Is $\hat{q}_L$ unbiased and consistent?

(c) Now compute $\text{MSE}_{\hat{q}}$. Not that in this case the bias is zero, so it should be straightforward to derive from the standard error. Surprisingly, $\text{MSE}_{\hat{q}} > \text{MSE}_{\hat{q}_L}$ for some values of N and q. For which values is this the case? Note that this is quite surprising, since it seems like $\hat{q}$ should be the best guess of q!

Exercise 8 (Impulse function features)**:** Consider the model

$$Y|X \sim \text{Normal}\left(\sum_{i=1}^{K} \beta_i \phi_i(X), \sigma^2\right)$$

Suppose that $X \in [0, 1)$ and define the intervals

$$I_i = \left[\frac{i-1}{K}, \frac{i}{K}\right)$$

Notice that

$$[0, 1) = I_1 \cup I_2 \cup \cdots \cup I_K.$$

That is, each x in $[0, 1)$ is in one of these disjoint intervals. Now introduce the features

$$\phi_i(x) = \begin{cases} 1 & x \in [(i-1)/K, i/K) \\ 0 & x \notin [(i-1)/K, i/K) \end{cases}$$

(a) Are ϕ_i orthogonal with respect to a random variable X taking values in $[0,1)$? Does it depend on the distribution of X?

(b) Using `statsmodels`, implement fitting the model with these features. You can make up your simulated data set to fit, or copy the code I used in class to fit the fourier and polynomial models. I recommend writing a function `phi(x,i)` which takes the array of predictors and outputs an array $[\phi_j(X_1),\dots,\phi_j(X_N)]$. Use $K = 10$ and $N = 100$.

(c) As usual let $\hat{\beta}_j$ be the fitted value of β_j using least squares, meaning the value that minimizes the squared residuals. Show that in this model $\hat{\beta}_j$ is simply the average value of Y_i among data points where $X_i \in I_j$; that is,

$$\hat{\beta}_j = \frac{1}{N_j} \sum_{i:X_i \in I_j} Y_i$$

where N_j is the number of points in I_j.