Chapter 2

Least Squares Regression

We have so far predicted salaries without taking into account any information about the person whose salary is being predicted. Of course, there are lots of things which can influence someone's salary: how much experience they have, their college GPA, what city they live in, and so forth. We call these things **features**. In this chapter, we'll see how to apply the framework of empirical risk minimization to the problem of predicting with multiple features.

2.1 Simple Linear Regression

As mentioned above, there are many features that could be useful in predicting someone's salary, but we will begin by considering just one: years of experience. That is, we wish to make an accurate prediction of a data scientist's salary by using only their experience as an input.

We believe that experience is related to pay. In particular, we have a feeling that the more experience someone has, the higher their salary is likely to be. This relationship is not *exact*, of course – there is no law that says that someone with x years of experience should be paid *exactly* some amount. But for the purposes of prediction, it is believable that there is some function $H(x)$ which takes in someone's experience level and outputs a prediction of their salary which is reasonably close to their actual salary.

We call such a function a **prediction rule**. You can think of it as a formula for making predictions. Here is one example of a prediction rule:

$$H_1(\text{years of experience}) = \$50{,}000 + \$2,000 \times (\text{years of experience})$$

This rule says that, to predict your pay, we should start with $50,000 and add $2,000 for every year of experience you have. In other words, pay increases **linearly** with experience. Another prediction rule is:

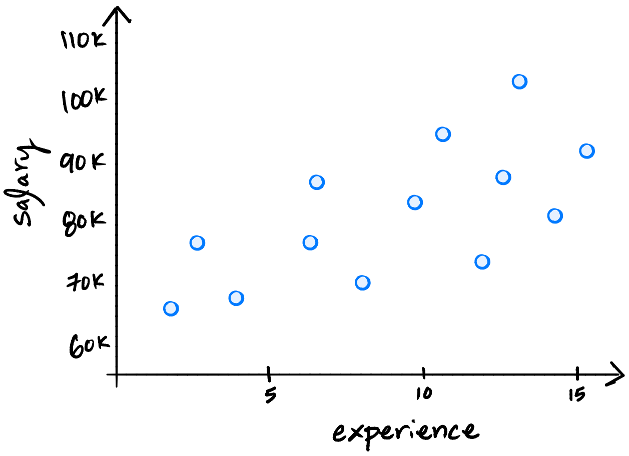
$$H_2(\text{years of experience}) = \$60{,}000 \times 1.05^{(\text{years of experience})}$$

In this rule, pay increases **exponentially** with experience. Yet another prediction rule is:

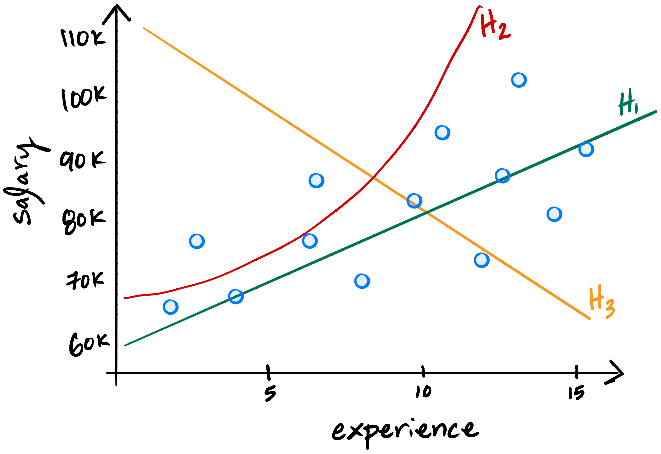
$$H_3(\text{years of experience}) = \$100{,}000 - \$5{,}000 \times (\text{years of experience})$$

This one says that your pay *decreases* with experience. This is (hopefully) not the case, and it goes against our experience. This is probably a *bad* prediction rule in the sense that it does not make accurate predictions, but it is a prediction rule nonetheless.

As before, we will assess whether a prediction rule is good or bad by seeing if the predictions it makes match the data. The figure below shows what we might expect to see if we ask several data scientists how much they make and how much experience they have:



In general, we observe a *positive* trend: the more experience someone has, the higher their pay tends to be. We can plot the decision rules H_1, H_2, and H_3 on top of this data to get a sense for the accuracy of their predictions:



Out of these, H_1 appears to "fit" the data the best. We'll now make this more precise.

2.1.1 Loss Functions

Suppose we have surveyed n data scientists, and for each recorded their experience, x_i, and their salary, y_i. We want to assess the accuracy of a prediction rule, $H(x)$, which takes as input years of experience and outputs the predicted salary.

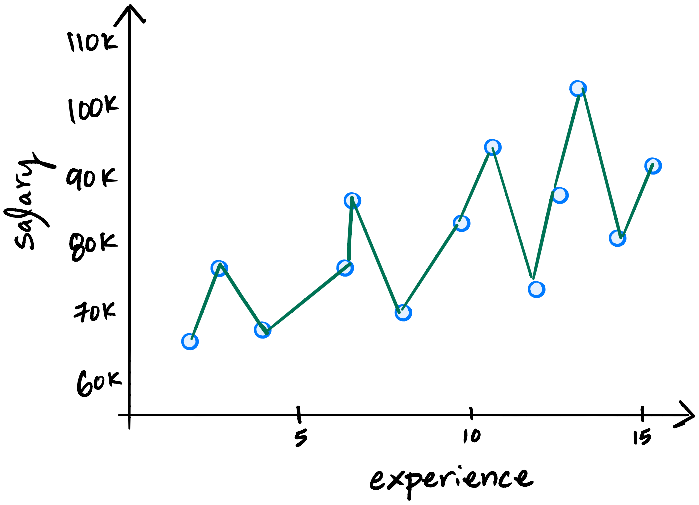
Consider an arbitrary person: person i. Their salary is y_i, and their experience is x_i. Our prediction for their salary is $H(x_i)$. The **absolute loss** of our prediction is

$$|H(x_i) - y_i|$$

In the last chapter, we saw that the *average loss* incurred when using H to predict the salary for everyone in the data set is one way to quantify how good or bad H is. In this case, the **mean absolute error** is:

$$R_{\text{abs}}(H) = \frac{1}{n} \sum_{i=1}^{n} |H(x_i) - y_i|$$

Our goal now is to find a prediction rule H which results in the smallest mean absolute error. Here we run into our first problem: it turns out that we can find a function which has *zero* absolute error, but it isn't as useful as we'd like! Here it is:



This decision rule makes exactly the right prediction for every person in the data set. But in order to go through every data point, this function has made itself quite wiggly. We don't believe that this wiggly of a function truly describes how pay is related to experience. To put it another way, we feel that the data has some noise, and the above function describes the noise rather than the underlying pattern that we are interested in. When this is the case, we say that the line has **overfit** the data.

One antidote to overfitting is to mandate that the hypothesis not be so wiggly; in other words, restrict the hypothesis space so that it doesn't include such complicated functions. This is an instance of Occam's Razor: we should choose the simplest possible description of the data which actually works. In this example, a straight line is a good, simple description of the data, and so we'll restrict the decision rule to be of the form $H(x) = w_1 x + w_0$, i.e., straight lines. This setting is called **simple linear regression**.

Our new goal is to find a *linear* decision rule with the smallest mean absolute error. That is, we want to solve:

$$H^* = \underset{\text{linear } H}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} |H(x_i) - y_i|$$

But there is still a problem with this: it turns out that R_{abs} is relatively hard to minimize.[1]

The cause of this difficulty is our use of the absolute loss. Instead, we'll try another approach to quantifying the error of our prediction rule: the **square loss**:

$$(H(x_i) - y_i)^2$$

[1] Although it can be done with linear programming.

With this, the average loss becomes:

$$R_{\text{abs}}(H) = \frac{1}{n}\sum_{i=1}^{n}(H(x_i) - y_i)^2$$

This is also known as the **mean squared error**. Minimizing this is often called **least squares regression**.

2.1.2 Minimizing the Mean Squared Error

Let's now minimize the mean squared error, $R_{\text{sq}}(h)$. Since R_{sq} is a function of H but H is determined by our choice of w_0 and w_1, we will also write $R_{\text{sq}}(h)$ as $R_{\text{sq}}(w_0, w_1)$ to indicate more clearly that the loss is determined by the values of w_0 and w_1. When we minimize the loss function, we are really trying to find the optimal values of w_0, w_1, those that define the best-fitting line for the data. We have:

$$\begin{aligned} R_{\text{sq}}(H) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - H(x_i))^2 \\ R_{\text{sq}}(w_0, w_1) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2 \\ R_{\text{sq}}(w_0, w_1) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2 \end{aligned}$$

This is differentiable. To find the best w_0 and w_1, we'll use calculus. Taking the partial derivatives with respect to w_0 and w_1 and setting them to zero gives:

$$\frac{\partial R_{\text{sq}}(w_0, w_1)}{\partial w_0} = 0 = \sum_{i=1}^{n} -2(y_i - w_0 - w_1 x_i) \tag{2.1}$$

$$\frac{\partial R_{\text{sq}}(w_0, w_1)}{\partial w_1} = 0 = \sum_{i=1}^{n} -2x_i(y_i - w_0 - w_1 x_i) \tag{2.2}$$

We must now solve this system of equations to find the optimal values of w_0 and w_1.

We will start by setting $\partial R_{\text{sq}} / \partial w_0 = 0$ and solving for w_0.[2]

[2] Note that we could have also decided to solve this equation for w_1, as long as we then solve the other equation for w_0.

$$\frac{\partial R_{\text{sq}}}{\partial w_0} = 0 \implies \frac{1}{n}\sum_{i=1}^{n} 2\left(w_1 x_i + w_0 - y_i\right) = 0$$

We can take the 2 outside of the sum:

$$\implies \frac{2}{n}\sum_{i=1}^{n} \left(w_1 x_i + w_0 - y_i\right) = 0$$

If we multiply both sides by $n/2$ we are left with a constant factor of 1 on the left hand side, whereas the right hand side remains zero. In effect, we can get rid of the $2/n$:

$$\implies \sum_{i=1}^{n} \left(w_1 x_i + w_0 - y_i\right) = 0$$

Our goal is to isolate the w_0 on one side of the equation, but we are momentarily prevented from this by the fact that w_0 is inside of the sum. We can remove it from the summation, however; the first step is to break it into three independent sums:

$$\begin{aligned}
&\implies \sum_{i=1}^{n} w_1 x_i + \sum_{i=1}^{n} w_0 - \sum_{i=1}^{n} y_i = 0 \\
&\implies \sum_{i=1}^{n} w_0 = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} w_1 x_i \\
&\implies n w_0 = \sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i \\
&\implies w_0 = \frac{1}{n}\left(\sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i\right)
\end{aligned}$$

Define $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$; if x_i and y_i are experience and salary, then $\bar{x}$ is the mean experience and $\bar{y}$ is the mean salary of people in our data set. Using this notation

$$\implies w_0 = \bar{y} - w_1 \bar{x}$$

And so we have successfully isolated w_0. We now solve $\partial R_{\text{sq}}/\partial w_1 = 0$ for w_1:

$$\frac{\partial R_{\text{sq}}}{\partial w_1} = 0 \implies \frac{1}{n}\sum_{i=1}^{n} 2\left(\left(w_1 x_i + w_0\right) - y_i\right) x_i = 0$$

We'll get rid of the 2 and the $1/n$ straight away:

$$\implies \sum_{i=1}^{n} \left((w_1 x_i + w_0) - y_i \right) x_i = 0$$

We now substitute our solution for w_0:

$$\implies \sum_{i=1}^{n} \left((w_1 x_i + \bar{y} - w_1 \bar{x}) - y_i \right) x_i = 0$$

We'll group the x_i with the $\bar{x}$ and the y_i with the $\bar{y}$:

$$\implies \sum_{i=1}^{n} \left((w_1 (x_i - \bar{x}) - (y_i - \bar{y}) \right) x_i = 0$$

Splitting the summand:

$$\begin{aligned}
&\implies \sum_{i=1}^{n} w_1 (x_i - \bar{x}) x_i - \sum_{i=1}^{n} (y_i - \bar{y}) x_i = 0 \\
&\implies w_1 \sum_{i=1}^{n} (x_i - \bar{x}) x_i = \sum_{i=1}^{n} (y_i - \bar{y}) x_i \\
&\implies w_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y}) x_i}{\sum_{i=1}^{n} (x_i - \bar{x}) x_i}
\end{aligned}$$

We could stop here; this is a totally valid formula for w_1. But we'll continue to get a formula with a little more symmetry. The key is that $\sum_{i=1}^{n} (y_i - \bar{y}) = 0$ and $\sum_{i=1}^{n} (x_i - \bar{x})$, as can be verified. This enables us to write:

$$\implies w_1 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

We'll show that the numerators of these last two formulas for w_1 are the same (and you can show that the denominators are, too). We have:

$$\begin{aligned}
\sum_{i=1}^{n} (y_i - \bar{y})(x_i - \bar{x}) &= \sum_{i=1}^{n} (y_i - \bar{y}) x_i - \sum_{i=1}^{n} (y_i - \bar{y}) \bar{x} \\
&= \sum_{i=1}^{n} (y_i - \bar{y}) x_i - \bar{x} \sum_{i=1}^{n} (y_i - \bar{y}) \\
&= \sum_{i=1}^{n} (y_i - \bar{y}) x_i - 0 \\
&= \sum_{i=1}^{n} (y_i - \bar{y}) x_i
\end{aligned}$$

We have arrived at a formula for w_1 (the slope of our hypothesis function) which involves only the data.

Therefore, given some data, we can compute w_1. On the other hand, our formula for the intercept, w_0, involves w_1. To calculate it, we plug in the value of w_1 that we have calculated from the data.

In summary, the formulas we have found are:

$$\boxed{\begin{aligned} w_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\ w_0 &= \bar{y} - w_1 \bar{x} \end{aligned}}$$

where $\bar{x}$ and $\bar{y}$ are the mean of the x_i and y_i, respectively. These are called the **lease squares solutions** for the slope and intercept parameters.

It should be emphasized that the fact that we have *formulas* which allow us to calculate w_0 and w_1 directly is a consequence of using the mean squared error. If we had used another loss function, for example, the mean absolute error, then we may not be so lucky to have formulas. Instead, we would have to minimize the loss using an algorithmic approach in which we iterate towards the correct answer.[3]

2.1.3 Fitting non-linear trends

You may be surprised to learn that the formulas we have just derived for fitting a straight line to the data can sometimes be cleverly used to to fit certain nonlinear curves to data. By applying a suitable transformation, it can be possible to turn a nonlinear relationship between x and y into a linear relationship between two quantities that can be directly computed from x and y.

For instance, We can fit a curve of the form

$$H(x) = w_0 + w_1 \ln x$$

to data with the tools we already have. While y does not vary linearly with x, it does vary linearly with $\ln x$. So if we let $z = \log x$ be our feature variable, we can use the estimators we have developed to find the slope and intercept of the linear relationship.

[3] There are even some losses which are so hard to optimize that we can't even hope to find an algorithm which does so exactly; we have to approximate their minimizers.

$$w_1 = \frac{\sum_{i=1}^{n}(\log x_i - \frac{1}{n}\sum_{i=1}^{n}\log x_i)(y_i - \bar{y})}{\sum_{i=1}^{n}(\log x_i - \frac{1}{n}\sum_{i=1}^{n}\log x_i)^2}$$

$$w_0 = \bar{y} - w_1 \cdot \frac{1}{n}\sum_{i=1}^{n}\log x_i$$

We can use w_1 and w_0 as the parameters that define the best-fitting curve of the form $y = w_0 + w_1 \ln x$.

In general, we can use the formulas we've derived to fit any prediction rule of the form $H(x) = w_1 \cdot f(x) + w_0$, where $f(x)$ is some transformation of x, such as x^2, e^x, $\log x$, and so on. Here is the procedure:

1. Create a new data set $(z_1, y_1), \ldots, (z_n, y_n)$, where $z_i = f(x_i)$.
2. Fit $H(z) = w_1 z + w_0$ using familiar least squares solutions:

   $$w_1 = \frac{\sum_{i=1}^{n}(z_i - \bar{z})(y_i - \bar{y})}{\sum_{i=1}^{n}(z_i - \bar{z})^2} \qquad w_0 = \bar{y} - w_1 \cdot \bar{z}$$

   where $\bar{z}$ is the mean of the z_i.
3. Use w_1 and w_0 in original decision rule, $H(x)$.

2.2 Multiple Linear Regression

Next, we will look at linear regression through the lens of linear algebra, which will eventually allow us to extend what we know to new settings. In particular, we'll be able to include more than one predictor variable in making our predictions.

We have defined the least squares line as the prediction rule $H(x) = w_0 + w_1 x$ for which the mean square error is minimized. That is, it is the line which minimizes the risk:

$$R_{\text{sq}}(H) = \frac{1}{n}\sum_{i=1}^{n}(y_i - H(x_i))^2$$

or, equivalently:

$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$$

Notice that the risk is a sum of squares (divided by n), and remember from linear algebra that we also measure the length of a vector using a sum of squares. For example, the length of the vector $\vec{x} = \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix}$ is computed as

$$\sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{2^2 + 5^2 + 4^2} = \sqrt{45} = \sqrt{9 \cdot 5} = 3\sqrt{5}$$

If we use $||\vec{v}||$ to denote the length of a vector $\vec{v}$, this means we can write $R_{\text{sq}}(w_0, w_1) = ||\vec{e}||^2$, where $\vec{e}$ is the vector whose ith component is $e_i = y_i - H(x_i)$, which is the error in the ith prediction. Since each component of $\vec{e}$ comes from a difference between y_i and $H(x_i)$, let's think of the vector $\vec{e}$ as a difference of two vectors $\vec{y}$ and $\vec{h}$, where the ith component of $\vec{y}$ is y_i and the ith component of $\vec{h}$ is $H(x_i)$. That is: $\vec{e} = \vec{y} - \vec{h}$, and the risk can be written as $R_{\text{sq}}(H) = \frac{1}{n}\|\vec{e}\|^2 = \frac{1}{n}\|\vec{y} - \vec{h}\|^2$.

Since our function $H(x)$ is a linear function of the form $H(x) = w_0 + w_1 x$, this means we can write the vector $\vec{h}$ as follows:

$$\vec{h} = \begin{bmatrix} H(x_1) \\ H(x_2) \\ \vdots \\ H(x_n) \end{bmatrix} = \begin{bmatrix} w_0 + w_1 x_1 \\ w_0 + w_1 x_2 \\ \vdots \\ w_0 + w_1 x_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}.$$

Letting $X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$ and $\vec{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$, we have $\vec{h} = X\vec{w}$.

Notice that we are able to write $\vec{h}$ in matrix form because our choice of hypothesis $H(x)$ was linear in the parameters. We'll soon be able to adjust the form of $H(x)$ to fit any model that is linear in the parameters, including arbitrary polynomials.

For now, we continue with our linear fit $H(x) = w_0 + w_1 x$. We have established that

$$R_{\text{sq}}(H) = R_{\text{sq}}(w_0, w_1) = \|\vec{y} - \vec{h}\|^2 = \|\vec{y} - X\vec{b}\|^2$$

The matrix X is called the design matrix, whereas the vector $\vec{w}$ is called the parameter vector, and the vector $\vec{y}$ is called the observation vector. Note that we can think of the loss function as a function of the parameter vector $\vec{w}$, as changing this parameter vector produces different amounts of loss, and X and y are completely determined by the data set. Our goal is to find the choice of $\vec{w}$ for which $R_{\text{sq}}(\vec{w}) = ||\vec{y} - X\vec{w}||^2$ is minimized. Equivalently, our goal is to find the vector $\vec{w}$ for which the vector

$\vec{y} - X\vec{w}$ has the smallest length, since minimizing the square of the length is equivalent to minimizing the length itself.

The smallest possible length of a vector is zero, since length is always nonnegative. What does it mean about the data set of pairs (x_i, y_i) in the case that $R_{\text{sq}}(\vec{w})$ has a minimum value of zero? Why?

Most of the time, we will be in the situation where $R_{\text{sq}}(\vec{w})$ has a minimum value greater than zero. We'll use calculus to find the value of $\vec{w}$ where the minimum is achieved.

2.2.1 Minimizing the Mean Squared Error

At this point, we have an expression for the mean squared error in matrix notation:

$$R_{\text{sq}}(\vec{w}) = \frac{1}{n}\|X\vec{w} - \vec{y}\|^2$$

Given the data, X and $\vec{y}$ are fixed, known quantities. Our goal is to find the "best" $\vec{w}$, in the sense that $\vec{w}$ results in the smallest mean squared error. We'll use vector calculus to do so.

Our main strategy for minimizing risk has been to take the derivative of the risk function, set to zero, and solve for the minimizer. In this case, we want to differentiate $R_{\text{sq}}(\vec{w})$ with respect to a *vector*, $\vec{w}$. The derivative of a scalar-valued function with respect to a vector input is called the **gradient**. Recall that the gradient $\frac{dR}{d\vec{w}}$ of $R(\vec{w})$ is itself a vector of partial derivatives:

$$\frac{dR}{d\vec{w}} = \begin{pmatrix} \frac{dR}{dw_1} \\ \frac{dR}{dw_2} \\ \vdots \\ \frac{dR}{dw_d} \end{pmatrix}$$

where $w_1, \ldots, w_d$ are the entries of the vector $\vec{w}$.

Our goal is to find the gradient of the mean squared error. We start by expanding the mean squared error in order to get rid of the squared norm. Recall that $\|\vec{u}\|^2 = \vec{u}^T\vec{u}$.

Therefore:

$$\begin{aligned}
R_{\text{sq}}(\vec{w}) &= \frac{1}{n}\|X\vec{w} - \vec{y}\|^2 \\
&= \frac{1}{n}\left(X\vec{w} - \vec{y}\right)^T\left(X\vec{w} - \vec{y}\right) \\
&= \frac{1}{n}\left(\vec{w}^T X^T - \vec{y}^T\right)\left(X\vec{w} - \vec{y}\right) \\
&= \frac{1}{n}\vec{w}^T X^T X\vec{w} - \vec{w}^T X^T\vec{y} - \vec{y}^T X\vec{w} + \vec{y}^T\vec{y}
\end{aligned}$$

Remember that $\vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}$ and observe that $\vec{w}^T X^T \vec{y}$ is the dot product of $\vec{y}$ with $\vec{w}^T X^T$. So $\vec{w}^T X^T \vec{y} = \vec{y}^T X\vec{w}$, and:

$$= \frac{1}{n}\left[\vec{w}^T X^T X\vec{w} - \vec{2}y^T X\vec{w} + \vec{y}^T\vec{y}\right]$$

Now we take the gradient. We have:

$$\begin{aligned}
\frac{dR_{\text{sq}}}{d\vec{w}} &= \frac{d}{d\vec{w}}\frac{1}{n}\left[\vec{w}^T X^T X\vec{w} - \vec{2}y^T X\vec{w} + \vec{y}^T\vec{y}\right] \\
&= \frac{1}{n}\left[\frac{d}{d\vec{w}}\left(\vec{w}^T X^T X\vec{w}\right) - \frac{d}{d\vec{w}}\left(\vec{2}y^T X\vec{w}\right) + \frac{d}{d\vec{w}}\left(\vec{y}^T\vec{y}\right)\right]
\end{aligned}$$

As shown in lecture, $\frac{d}{d\vec{w}}\left(\vec{w}^T X^T X\vec{w}\right) = 2X^T X\vec{w}$. Similarly, $\frac{d}{d\vec{w}}\left(\vec{y}^T X\vec{w}\right) = X^T\vec{y}$. And the gradient of $\vec{y}^T\vec{y}$ with respect to $\vec{w}$ is clearly zero, since $\vec{y}$ is a constant vector. Therefore:

$$= 2X^T X\vec{w} - 2X^T\vec{y}$$

This is the gradient of the mean squared error. Our next step is to set this equal to zero and solve for $\vec{w}$:

$$\begin{aligned}
\frac{dR_{\text{sq}}}{d\vec{w}} = 0 &\implies 2X^T X\vec{w} - 2X^T\vec{y} = 0 \\
&\implies X^T X\vec{w} = X^T\vec{y}
\end{aligned}$$

The last line is an important one. The equation $X^T X\vec{w} = X^T\vec{y}$ defines a system of equations in matrix form known as the **normal equations**. This system can be solved using Gaussian elimination, or, if the matrix $X^T X$ is invertible, by multiplying both sides by its inverse to obtain $\vec{w} = (X^T X)^{-1} X^T\vec{y}$.

The normal equations, $X^T X\vec{w} = X^T\vec{y}$ are the solution to the least squares problem. Before, we had formulas for the least squares solutions for w_0 and w_1; the normal equations are equivalent. In fact, you can derive our old formulas from the normal equations with a little bit of algebra. The advantage of the normal equations will be that they generalize to the setting where we have not just one feature, but many.

Example: Find the linear function that best approximates the data $(2,1)$, $(5,2)$, $(7,3)$, $(8,3)$, where the first element of each pair is the feature x, and the second element is the output, y.

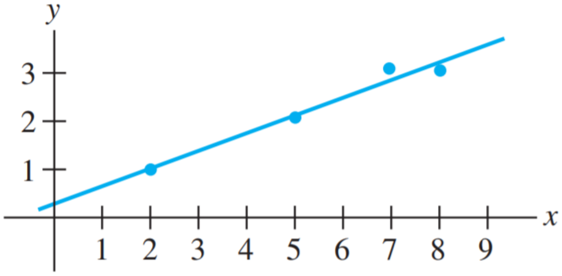
We have $\vec{y} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 3 \end{bmatrix}$ and $X = \begin{bmatrix} 1 & 2 \\ 1 & 5 \\ 1 & 7 \\ 1 & 8 \end{bmatrix}$.

We can calculate $X^T X = \begin{bmatrix} 4 & 22 \\ 22 & 142 \end{bmatrix}$ and $X^T y = \begin{bmatrix} 9 \\ 57 \end{bmatrix}$, so the normal equations are

$$\begin{bmatrix} 4 & 22 \\ 22 & 142 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} 9 \\ 57 \end{bmatrix}.$$

Now, we just need to solve this linear system. The left-hand side matrix can be inverted and both sides multiplied by it to find $\vec{w} = \begin{bmatrix} \frac{2}{7} \\ \frac{5}{14} \end{bmatrix}$.

The best fit line is therefore $y = \frac{2}{7} + \frac{5}{14}x$, as shown in the picture below.

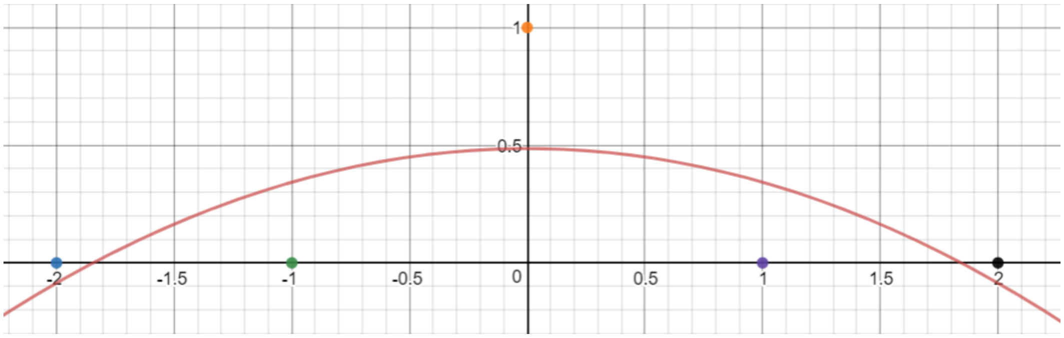


2.2.2 Using Multiple Features

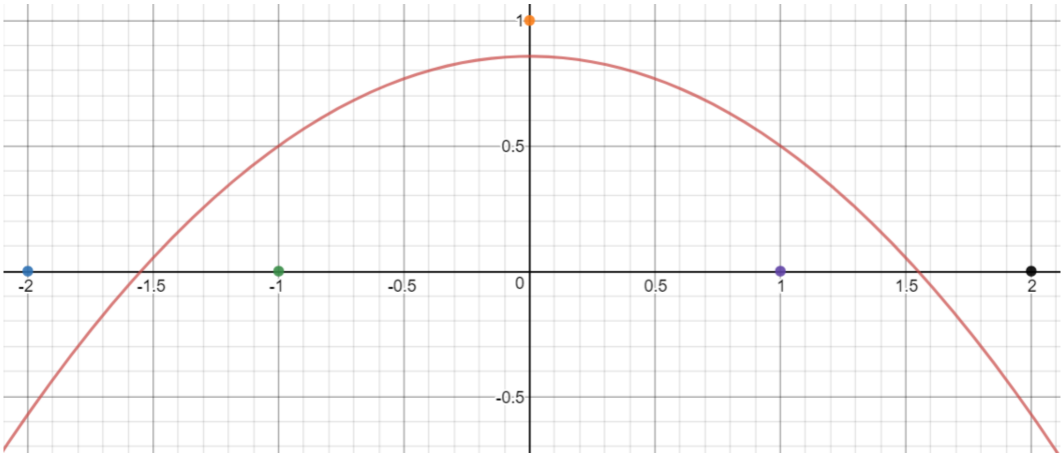
The really nice thing about this linear algebra approach to least squares is that we can use the same strategy to fit many different curves to a set of data points. Before, we only developed estimators for the slope and intercept of the regression *line*; we did not have the tools to, say, fit an arbitary quadratic of the form $y = c_0 + c_1 x + c_2 x^2$. With the linear algebra method, we can now do this. For example, consider the set of points $(-2,0),(-1,0),(0,1),(1,0),(2,0)$. The system of equations corresponding to a parabola that goes through these points has design matrix

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_5 & x_5^2 \end{bmatrix} = \begin{bmatrix} 1 & -2 & 4 \\ 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{bmatrix}$$

parameter vector $\vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$ and observation vector $y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$. We can calculate that $X^TX = \begin{bmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix}$ and $X^Ty = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ so that the least squares solution satisfies $\begin{bmatrix} 5 & 0 & 10 \\ 0 & 10 & 0 \\ 10 & 0 & 34 \end{bmatrix} \cdot \vec{w} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$. Solving this system of equations gives $\vec{w} = \begin{bmatrix} 34/70 \\ 0 \\ -10/70 \end{bmatrix}$ so that the best-fitting quadratic is $y = \frac{34}{70} - \frac{10}{70}x^2$. The plot below shows the data points and the best fitting quadratic.



Compare the fit of the quadratic above with the fit of a different quadratic below by drawing in squares whose area you are trying to minimize. You should find that the the area of the squares is smaller in the top picture than in the bottom picture. Since the top picture represents the *least* squares solution, the total area of the squares will be as small as possible among all quadratics.



The least squares method we have developed using linear algebra can be used to find the best-fitting curve of any function form that is linear in the parameters. This includes all polynomials, because any polynomial

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$$

is linear in the coefficients $c_0, c_1, c_2, \ldots, c_n$.

The linear algebra approach is also useful for fitting a function of multiple variables to data, under the same stipulation that the function should be linear in the parameters. This is called *multiple regression* because we are using multiple predictor variables to predict the value of the response variable. This is highly useful because the more predictor variables we take into consideration when predicting the value of the response variable, the more likely we are to make an accurate prediction. For example, Galton might have been able to more accurate predictions had he used the mother's height and father's height as separate predictor variables, rather than lumping them together into a single midparent height.

For example, suppose we want to predict the price of a laptop computer based on its weight and amount of memory. We could collect data for existing computers, recording their price p, weight w, and memory m. Suppose that a plot of the data (w_i, m_i, p_i) shows that it makes sense to fit a curve of the form

$$p = c_0 + c_1 w + c_2 m,$$

which is the equation of a plane. Then we can use design matrix $X = \begin{bmatrix} 1 & w_1 & m_1 \\ 1 & w_2 & m_2 \\ \vdots & \vdots & \vdots \\ 1 & w_n & m_n \end{bmatrix}$, parameter vector $\vec{w} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$ and observation vector $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$, and find the equation of the least squares plane by solving $X^T X\vec{w} = X^T\vec{y}$. If you collected actual data and used this approach to find the values of the parameters, what signs would you expect the parameters c_1 and c_2 to have? The more memory the computer has, the more expensive it probably is, and so c_2 would be positive. As for c_1, you could make the case either way: people are willing to spend extra for ultralight laptops, but at the same time, smaller, cheaper computers tend to weigh less than expensive and large gaming laptops with bigger screens. So it isn't clear whether c_1 will be positive or negative.

Of course, the more predictor variables we have, the better predictions we are able to make, right? Not necessarily. There is such a thing as having *too many* variables. Suppose we are trying to predict a child's height. To do so, we gather as many predictor variables as possible. Some predictors, like the height of each parent, are clearly related to the outcome. But other variables, like the day of the week on which the child was born, are probably unrelated. However, if we have too many of these unrelated variables, by sheer *chance* a pattern can emerge. This pattern isn't meaningful – it's just due to noise – but least squares regression does not know the difference, and will happily *overfit* the noise.

More predictor variables also introduce other challenges. For one, it is very hard to visualize multidimensional data, especially data in more than three dimensions, since we don't have a good way of plotting such data. Then it can be hard to guess the appropriate form of the function to fit. Even if we can come up with an appropriate function form, actually solving the matrix equation can be computationally difficult if the matrix is very large.