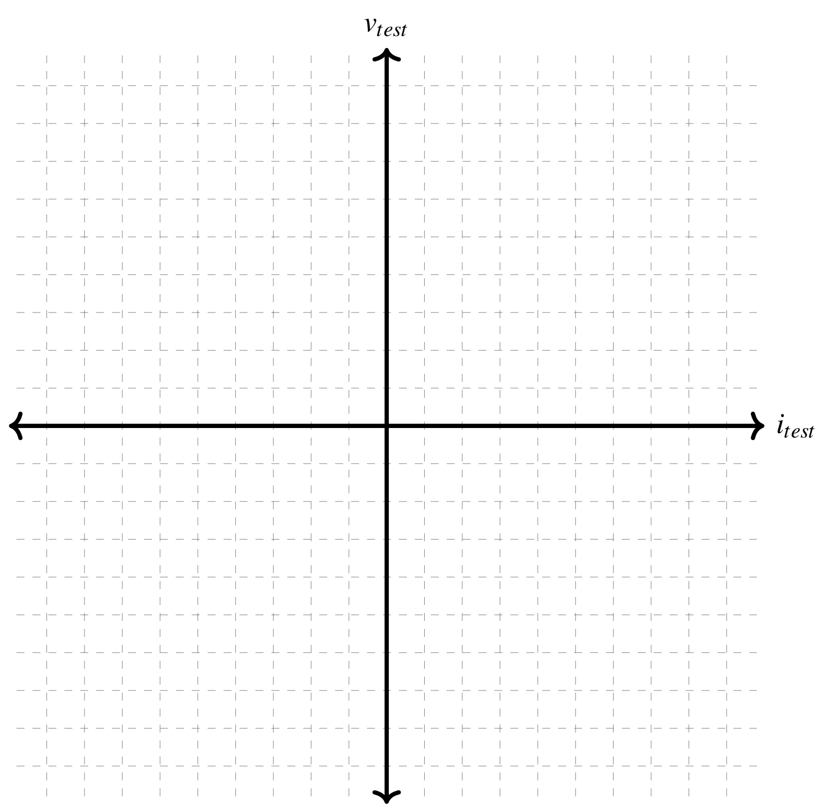
EECS 16A Designing Information Devices and Systems I
Summer 2023 Discussion 07D

1. Ohm's Law With Noise

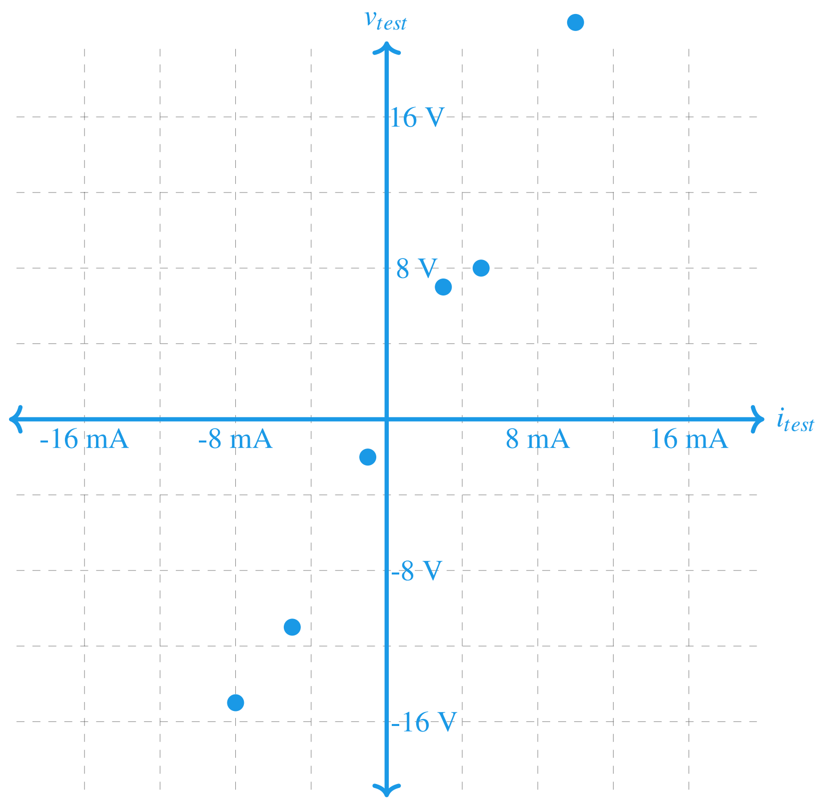
We are trying to measure the resistance of a black box. We apply various i_{test} currents and measure the ouput voltage v_{test}. Oftentimes, our measurement tools are a little bit noisy, and the values we get out of them are not accurate. However, if the noise is completely random, then it turns out that if we gather many measurements and use them all to find the best solution, the effect of the noise can be averaged out. So we repeat our test many times:

Test	i_{test} (mA)	v_{test} (V)
1	10	21
2	3	7
3	-1	-2
4	5	8
5	-8	-15
6	-5	-11

(a) Plot the measured voltage as a function of the current.



Answer:



Notice that these points *do not* lie on a line!

(b) Suppose we stack the currents and voltages to get $\vec{I} = \begin{bmatrix} 10 \\ 3 \\ -1 \\ 5 \\ -8 \\ -5 \end{bmatrix}$ and $\vec{V} = \begin{bmatrix} 21 \\ 7 \\ -2 \\ 8 \\ -15 \\ -11 \end{bmatrix}$. Is there a unique solution for R? What conditions must $\vec{I}$ and $\vec{V}$ satisfy in order for us to solve for R uniquely?

Answer:

We cannot find the unique solution for R because $\vec{V}$ is not a scalar multiple of $\vec{I}$. In general, we need $\vec{V}$ to be a scalar multiple of $\vec{I}$ to be able to solve for R exactly (another linear algebraic way of saying this is that $\vec{V}$ is in the span of $\vec{I}$).

We know that the *physical* reason we are not able to solve for R is that we have imperfect observations of the voltage across the terminals, $\vec{V}$. Therefore, now that we know we cannot solve for R directly, a very pertinent goal would be to find a value of R that *approximates* the relationship between $\vec{I}$ and $\vec{V}$ as closely as possible.

Let's move on and see how we do this.

(c) Ideally, we would like to find R such that $\vec{V} = \vec{I}R$. If we cannot do this, we'd like to find a value of R that is the *best* solution possible, in the sense that $\vec{I}R$ is as "close" to $\vec{V}$ as possible. We are defining the sum of squared errors as a **cost function**. In this case the cost function for any value of R quantifies the difference between each component of $\vec{V}$ (i.e. v_j) and each component of $\vec{I}R$ (i.e. i_jR) and sum up the squares of these "differences" as follows:

$$\text{cost}(R) = \sum_{j=1}^{6} (v_j - i_j R)^2$$

Do you think this is a good cost function? Why or why not?

Answer:

For each point (i_j, v_j), we want $|v_j - i_j R|$ to be as small as possible. We can call this term the individual error term for this point.

One way of looking at the aggregate "error" in our fit is to add up the squares of the individual errors, so that all errors add up. This is precisely what we've done in the cost function. If we did not square the differences, then a positive difference and a negative difference would cancel each other out.

(d) Show that you can also express the above cost function in vector form, that is,

$$\text{cost}(R) = \left\langle (\vec{V} - \vec{I}R), (\vec{V} - \vec{I}R) \right\rangle$$

Hint: $\langle \vec{a}, \vec{b} \rangle = \vec{a}^T \vec{b} = \sum_i a_i b_i$

Answer:

Let's define the error vector as

$$\vec{e} = \vec{V} - \vec{I}R.$$

Then, we observe that $e_j = v_j - i_j R$.

Therefore,

$$\begin{aligned}
\text{cost}(R) &= \sum_{j=1}^{6} (v_j - i_j R)^2 \\
&= \sum_{j=1}^{6} e_j^2 \\
&= \|\vec{e}\|_2^2 \\
&= \langle \vec{e}, \vec{e} \rangle \\
&= \left\langle (\vec{V} - \vec{I}R), (\vec{V} - \vec{I}R) \right\rangle
\end{aligned}$$

(e) Find $\hat{R}$, which is defined as the optimal value of R that minimizes $\text{cost}(R)$.

Hint: Use calculus. The optimal $\hat{R}$ makes $\frac{d\text{cost}(\hat{R})}{dR} = 0$

Answer: Recall from calculus that at the local minima and maxima of a function, the derivative of the function is equal to zero. It turns out that the least squares cost function only has one local extrema, and it is the global minimum of the function, so we will set the derivative to zero to find that minimum. First, let us write out the derivative:

$$\frac{d\text{cost}(R)}{dR} = -2 \sum_{j=1}^{6} i_j (v_j - i_j R)$$

It should be zero at $R = \hat{R}$, which means that we can solve for $\hat{R}$:

$$-2 \sum_{j=1}^{6} i_j (v_j - i_j \hat{R}) = 0,$$

After some algebra, we will get:

$$\hat{R} = \frac{\sum_{j=1}^{6} i_j v_j}{\sum_{j=1}^{6} i_j^2} = \frac{\left\langle \vec{I}, \vec{V} \right\rangle}{\|\vec{I}\|^2}$$

In our particular example, $\left\langle \vec{I}, \vec{V} \right\rangle = 448$ and $\|\vec{I}\|^2 = 224$. Therefore, $\hat{R} = 2\,\mathrm{k}\Omega$.

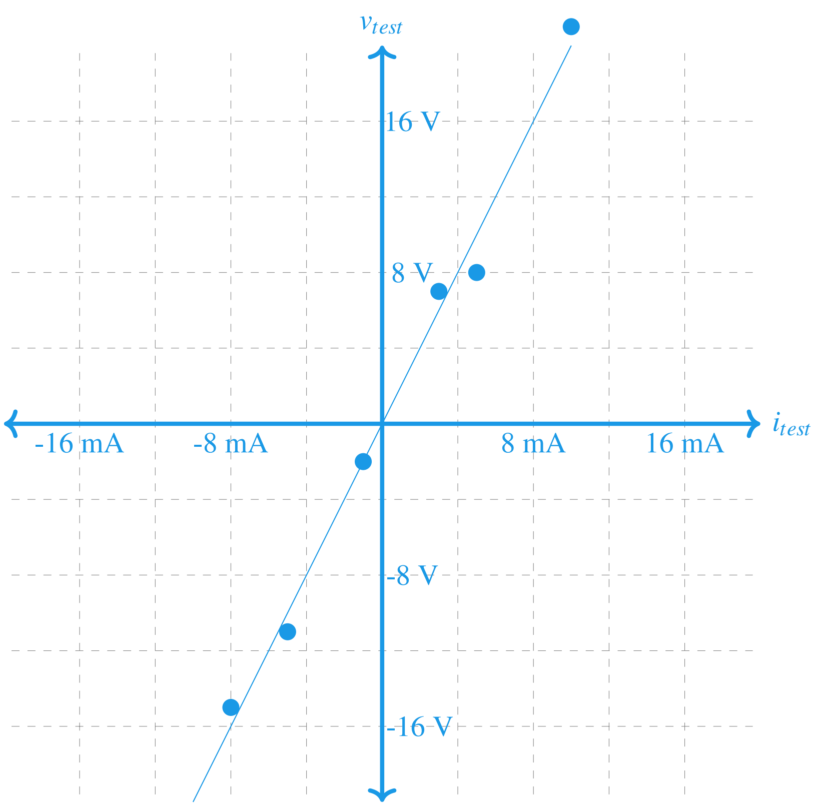
We will now see that if we use the equation for the optimal least squares estimate, we will get the same answer. Let $A = \vec{I}$ and $\vec{b} = \vec{V}$. Then,

$$\begin{aligned}
\hat{R} &= (A^T A)^{-1} A^T \vec{b} \\
&= (\,\vec{I}^T \vec{I}\,)^{-1} \vec{I}^T \vec{V} \\
&= \frac{\left\langle \vec{I}, \vec{V} \right\rangle}{\left\langle \vec{I}, \vec{I} \right\rangle} \\
&= \frac{\left\langle \vec{I}, \vec{V} \right\rangle}{\|\vec{I}\|^2},
\end{aligned}$$

which gives us the same expression as before! Note that because A is a vector in this case, $A^T A$ is a scalar, so to take its inverse we can just take a scalar inverse.

(f) On your original IV plot, also plot the line $v_{test} = \hat{R} i_{test}$. Can you visually see why this line "fits" the data well? How well would we have done if we had guessed $R = 3\,\mathrm{k}\Omega$? What about $R = 1\,\mathrm{k}\Omega$? Calculate the cost functions for each of these choices of R to validate your answer.

Answer:



When $\hat{R} = 2\,\text{k}\Omega$, we have

$$\begin{aligned}\text{cost}(2k) &= (21 - 2 \cdot 10)^2 + (7 - 2 \cdot 3)^2 + (-2 - 2 \cdot (-1))^2 + \\ &\quad (8 - 2 \cdot 5)^2 + (-15 - 2 \cdot (-8))^2 + (-11 - 2 \cdot (-5))^2 \\ &= 8.\end{aligned}$$

When $\hat{R} = 3\,\text{k}\Omega$, we have

$$\begin{aligned}\text{cost}(3k) &= (21 - 3 \cdot 10)^2 + (7 - 3 \cdot 3)^2 + (-2 - 3 \cdot (-1))^2 + \\ &\quad (8 - 3 \cdot 5)^2 + (-15 - 3 \cdot (-8))^2 + (-11 - 3 \cdot (-5))^2 \\ &= 232.\end{aligned}$$

When $\hat{R} = 1\,\text{k}\Omega$, we have

$$\begin{aligned}\text{cost}(1k) &= (21 - 1 \cdot 10)^2 + (7 - 1 \cdot 3)^2 + (-2 - 1 \cdot (-1))^2 + \\ &\quad (8 - 1 \cdot 5)^2 + (-15 - 1 \cdot (-8))^2 + (-11 - 1 \cdot (-5))^2 \\ &= 232.\end{aligned}$$

Notice that as expected, the least squares solution provides by far the best fit to the data, which you can see both visually and by the value of the cost function.

(g) Now, suppose that we add a new data point: $i_7 = 2\,\text{mA}$, $v_7 = 4\,\text{V}$. Will $\hat{R}$ increase, decrease, or remain the same? Why?

Answer: Notice that this new data point lies exactly on the previous best fit line $v_{test} = \hat{R} i_{test}$. Each data point in a least squares problem pushes the best fit line a little bit towards that point, but since the new point is already on the line, there is no reason for the line to change!

2. Polynomial Fitting

Even though least squares can only be applied to linear systems, it turns out that it can also solve problems with decidedly nonlinear elements. In lecture, you will see an example of fitting to an ellipse. Here, we will fit to a nonlinear polynomial.

Say we know that y is a quartic polynomial in x. This means that we know that y and x are related as follows:

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$$

We're also given the following observations, and our goal is to figure out the relationship between x and y:

x	y
0.0	24.0
0.5	6.61
1.0	0.0
1.5	−0.95
2.0	0.07
2.5	0.73
3.0	−0.12
3.5	−0.83
4.0	−0.04
4.5	6.42

(a) What are the unknowns in this question?

Answer:

The unknowns are a_0, a_1, a_2, a_3, and a_4.

(b) Can you write an equation corresponding to the first observation (x_0, y_0), in terms of a_0, a_1, a_2, a_3, and a_4? What does this equation look like? Is it linear in the unknowns?

Answer:

Plugging (x_0, y_0) into the expression for y in terms of x, we get

$$24 = a_0 + a_1 \cdot 0 + a_2 \cdot 0^2 + a_3 \cdot 0^3 + a_4 \cdot 0^4$$

You can see that this equation is linear in a_0, a_1, a_2, a_3, and a_4.

(c) Now, write a system of equations in terms of a_0, a_1, a_2, a_3, and a_4 using *all of the observations*.

Answer:

Write the next equation using the second observation. You will now get:

$$6.61 = a_0 + a_1 \cdot (0.5) + a_2 \cdot (0.5)^2 + a_3 \cdot (0.5)^3 + a_4 \cdot (0.5)^4$$

And for the third:

$$0.0 = a_0 + a_1 \cdot (1) + a_2 \cdot 1^2 + a_3 \cdot 1^3 + a_4 \cdot 1^4$$

Do you see a pattern? Let's write the entire system of equations in terms of a matrix now.

$$\begin{bmatrix} 1 & 0 & 0^2 & 0^3 & 0^4 \\ 1 & 0.5 & (0.5)^2 & (0.5)^3 & (0.5)^4 \\ 1 & 1 & 1^2 & 1^3 & 1^4 \\ 1 & 1.5 & (1.5)^2 & (1.5)^3 & (1.5)^4 \\ 1 & 2 & 2^2 & 2^3 & 2^4 \\ 1 & 2.5 & (2.5)^2 & (2.5)^3 & (2.5)^4 \\ 1 & 3 & 3^2 & 3^3 & 3^4 \\ 1 & 3.5 & (3.5)^2 & (3.5)^3 & (3.5)^4 \\ 1 & 4 & 4^2 & 4^3 & 4^4 \\ 1 & 4.5 & (4.5)^2 & (4.5)^3 & (4.5)^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} 24 \\ 6.61 \\ 0.0 \\ -0.95 \\ 0.07 \\ 0.73 \\ -0.12 \\ -0.83 \\ -0.04 \\ 6.42 \end{bmatrix}$$

(d) Finally, solve for a_0, a_1, a_2, a_3, and a_4 using IPython or any method you like. You have now found the quartic polynomial that best fits the data!

Answer:

Let $\mathbf{D}$ be the big matrix from the previous part.

$$\vec{a} = (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T\vec{y} = \begin{bmatrix} 24.00958042 \\ -49.99515152 \\ 35.0039627 \\ -9.99561772 \\ 0.99841492 \end{bmatrix}$$

It turns out that the actual parameters for the polynomial equation were:

$$\vec{a} = \begin{bmatrix} 24 \\ -50 \\ 35 \\ -10 \\ 1 \end{bmatrix}$$

(Remember that our observations were noisy.)

Therefore, we have actually done pretty well with the least squares estimate!

(e) What if we didn't know the degree of the polynomial? Use the IPython Notebook to explore what happens when we choose a polynomial degree other 4 and explain what you see.

Answer: You'll notice that if you try to fit the data with a polynomial of degree < 4, your solution will be a worse fit than the quartic fit we got before. This makes sense; you're basically trying to fit the data with a *smaller set of parameters*, which means that you are trying to approximate the data by a coarser model. It is only natural that you are going to do worse, and our cost function tells us so. Now, what if you tried to fit the data with a polynomial of degree > 4? Your cost function will improve but ever so slightly. You're not gaining much, and furthermore, you are *overfitting* your data after some point of time. When you are overfitting, your model starts to describe the underlying noise in the data rather than the data itself. This means that it is actually a worse model for making future predictions! This is called the bias variance theorem in statistics and traditional ML.

(f) OPTIONAL: Play around with what happens when you add more noise to the data or if you decide to drop data points on the IPython Notebook. Additionally, explore what you see when you change the degree of the polynomial alongside these factors.

Answer: You'll notice that if you increase the noise or you remove more data points, the polynomial fit becomes worse compared to the original data, with the errors becoming increasingly larger. However, if you play around with the degree of the polynomial, you might notice that if we choose our polynomial to be around degree 4-6, when we drop a few points or increase the noise moderately, it still gives us a polynomial fit that is very close to the original. If our degree was 10-15, dropping points or increasing our noise would wildly change our polynomial. This is another downside of overfitting our model, it is extremely susceptible to noisy or missing data.