
EECS 16A Designing Information Devices and Systems I

Fall 2021

Homework 13

This homework is due December 3, 2021, at 23:59.

Self-grades are due December 6, 2021, at 23:59.

Submission Format

Your homework submission should consist of **one** file.

- `hw13.pdf`: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook saved as a PDF. If you do not attach a PDF “printout” of your IPython notebook, you will not receive credit for problems that involve coding. Make sure that your results and your plots are visible. Assign the IPython printout to the correct problem(s) on Gradescope.

Submit the file to the appropriate assignment on Gradescope.

1. Reading Assignment

For this homework, please read Note 22 (Trilateration and Correlation) and Note 23 (Least Squares).

2. Audio File Matching

Learning Goal: *This problem motivates the application of correlation for pattern matching applications such as Shazam.*

Many audio processing applications rely on representing audio files as vectors, referred to as audio *signals*. Every component of the vector determines the sound we hear at a given time. We can use inner products to determine if a particular audio clip is part of a longer song, similar to an application like *Shazam*.

Let us consider a very simplified model for an audio signal, \vec{x} . At each timestep k , the audio signal can be either $x[k] = -1$ or $x[k] = 1$.

- (a) Say we want to compare two audio files of the same length N to decide how similar they are. First, consider two vectors that are exactly identical, namely $\vec{x}_1 = [1 \ 1 \ \dots \ 1]^T$ and $\vec{x}_2 = [1 \ 1 \ \dots \ 1]^T$. What is the inner product of these two vectors? What if $\vec{x}_1 = [1 \ 1 \ \dots \ 1]^T$ but \vec{x}_2 oscillates between 1 and -1 ? Assume that N , the length of the two vectors, is an even number. Use this to suggest a method for comparing the similarity between a generic pair of length- N vectors.

- (b) Next, suppose we want to find a short audio clip in a longer one. We might want to do this for an application like *Shazam*, which is able to identify a song from a short clip. Consider the vector of length 8, $\vec{x} = [-1 \ 1 \ 1 \ -1 \ 1 \ 1 \ -1 \ 1]^T$.

We want to find the short segment $\vec{y} := [y[0] \ y[1] \ y[2]]^T = [1 \ 1 \ -1]^T$ in the longer vector. To do this, perform the linear cross correlation between these two finite length sequences and identify at what shift(s) the linear cross correlation is maximized. Apply the same technique to identify what shift(s) gives the best match for $\vec{y} = [1 \ 1 \ 1]^T$.

(If you wish, you may use iPython to do this part of the question, but you do not have to.)

- (c) Now suppose our audio vector is represented using integers beyond simply just 1 and -1 . Find the short audio clip $\vec{y} = [1 \ 2 \ 3]^T$ in the song given by $\vec{x} = [1 \ 2 \ 3 \ 1 \ 2 \ 2 \ 3 \ 10]^T$. Where do you expect to see the peak in the correlation of the two signals? Is the peak where you want it to be, i.e. does it pull out the clip of the song that you intended? Why?
(If you wish, you may use iPython to do this part of the question, but you do not have to.)
- (d) Let us think about how to get around the issue in the previous part. We applied cross-correlation to compare segments of \vec{x} of length 3 (which is the length of \vec{y}) with \vec{y} . Instead of directly taking the cross correlation, we want to normalize each inner product computed at each shift by the magnitudes of both segments, i.e. we want to consider the inner product $\langle \frac{\vec{x}_k}{\|\vec{x}_k\|}, \frac{\vec{y}}{\|\vec{y}\|} \rangle$, where \vec{x}_k is the length 3 segment starting from the k -th index of \vec{x} . This is referred to as normalized cross correlation. Using this procedure, now which segment matches the short audio clip best?
- (e) We can use this on a more ‘realistic’ audio signal – refer to the IPython notebook, where we use normalized cross-correlation on a real song. Run the cells to listen to the song we are searching through, and add a simple comparison function `vector_compare` to find where in the song the clip comes from. Running this may take a couple minutes on your machine, but note that this computation can be highly optimized and run super fast in the real world! Also note that this is not exactly how Shazam works, but it draws heavily on some of these basic ideas.

3. Mechanical Trilateration

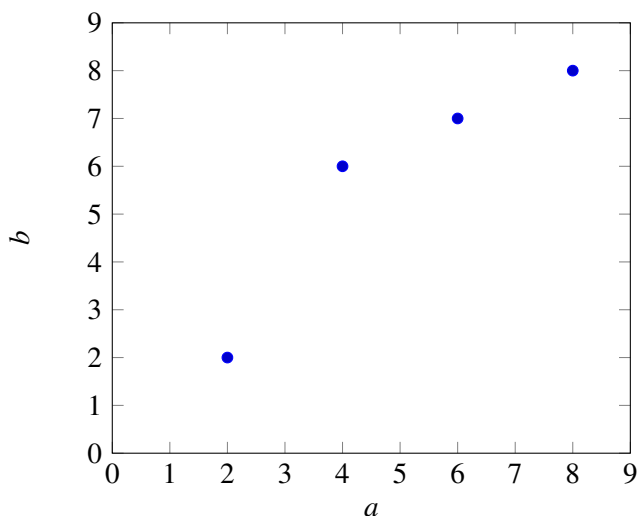
Learning Goal: The objective of this problem is to practice using trilateration to find the position based on the distance measurements and known beacon locations.

Trilateration is the problem of finding one’s coordinates given distances from known beacon locations. For each of the following trilateration problems, you are given 3 beacon locations ($\vec{s}_1, \vec{s}_2, \vec{s}_3$) and the corresponding distance (d_1, d_2, d_3) from each beacon to your location.

- (a) $\vec{s}_1 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$, $d_1 = 5$, $\vec{s}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, $d_2 = 2$, $\vec{s}_3 = \begin{bmatrix} -11 \\ 6 \end{bmatrix}$, $d_3 = 13$. First, use any graphing calculator or ipython to graph the set of constraints given by $(\vec{s}_1, \vec{s}_2, \vec{s}_3)$ and (d_1, d_2, d_3) , and take note of the number of solutions, or possible locations that you could be. Then use trilateration to find your location or possible locations. If a solution does not exist, state that it does not.
- (b) $\vec{s}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $d_1 = 5\sqrt{2}$, $\vec{s}_2 = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$, $d_2 = 5\sqrt{2}$, $\vec{s}_3 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$, $d_3 = 5$. First, use any graphing calculator or ipython to graph the set of constraints given by $(\vec{s}_1, \vec{s}_2, \vec{s}_3)$ and (d_1, d_2, d_3) , and take note of the number of solutions, or possible locations that you could be. Then use trilateration to find your location or possible locations. Why can’t we precisely determine our location, even though we have the same number of measurements as part (a)? Can we use our original constraints to narrow down our set of possible solutions we got from trilateration?
- (c) $\vec{s}_1 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$, $d_1 = 5$, $\vec{s}_2 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}$, $d_2 = 2$, $\vec{s}_3 = \begin{bmatrix} -12 \\ 5 \end{bmatrix}$, $d_3 = 12$. First, use any graphing calculator or ipython to graph the set of constraints given by $(\vec{s}_1, \vec{s}_2, \vec{s}_3)$ and (d_1, d_2, d_3) , and take note of the number of solutions, or possible locations that you could be. Then use trilateration to find your location or possible locations. If a solution does not exist, state that it does not.

Note that the next few problems will require knowledge of Tuesday’s (11/23) lecture.

4. Mechanical: Linear Least Squares



a	2	4	6	8
b	2	6	7	8

- (a) Consider the above data points. Find the linear model of the form

$$\vec{b} = \vec{a}x$$

that best fits the data, i.e. find the scalar value of x that minimizes the squared error

$$\|\vec{e}\|^2 = \left\| \begin{bmatrix} b_1 \\ \vdots \\ b_4 \end{bmatrix} - \begin{bmatrix} a_1 \\ \vdots \\ a_4 \end{bmatrix} x \right\|^2 = \|\vec{b} - \vec{a}x\|^2. \quad (1)$$

Note: by using this linear model, we are implicitly forcing the fit equation to go through the origin, i.e. $0 = x0$ for all x .

Do not use IPython for this calculation and show your work. Once you've computed x , compute the squared error between your model's prediction and the actual \vec{b} values as shown in Equation 1. Plot the best fit line along with the data points to examine the quality of the fit. (It is okay if your plot of $\vec{b} = \vec{a}x$ is approximate.)

Reminder: $\hat{x} = (\vec{a}^T \vec{a})^{-1} \vec{a}^T \vec{b}$

- (b) You will notice from your graph that you can get a better fit by adding a b -intercept. That is we can get a better fit for the data by assuming a linear model of the form

$$\vec{b} = x_1 \vec{a} + x_2.$$

In order to do this, we need to augment our \mathbf{A} matrix for the least squares calculation with a column of 1's (do you see why?), so that it has the form

$$\mathbf{A} = \begin{bmatrix} a_1 & 1 \\ \vdots & \vdots \\ a_4 & 1 \end{bmatrix}.$$

Find x_1 and x_2 that minimize the squared error

$$\|\vec{e}\|^2 = \left\| \begin{bmatrix} b_1 \\ \vdots \\ b_4 \end{bmatrix} - \begin{bmatrix} a_1 & 1 \\ \vdots & \vdots \\ a_4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|^2. \quad (2)$$

Do not use IPython for this calculation and show your work.

Reminder: $\hat{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \vec{b}$

Reminder: $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

Compute the squared error between your model's prediction and the actual \vec{b} values as shown in Equation 2. Plot your new linear model. Is it a better fit for the data?

5. Proof: Least Squares

Let \vec{x} be the solution to a linear least squares problem.

$$\vec{x} = \operatorname{argmin}_{\vec{x}} \|\vec{b} - \mathbf{A}\vec{x}\|^2$$

Show that the error vector $\vec{b} - \mathbf{A}\vec{x}$ is orthogonal to the columns of \mathbf{A} by direct manipulation (i.e. plug the formula for the linear least squares estimate into the error vector and then check if \mathbf{A}^T times the vector is the zero vector.)

6. How Much Is Too Much?

When discussing circuits in this course, we only talked about resistor I - V curves. There are many other devices that can be found in nature that do not have linear I - V relations. Instead, I is some general function of V , that is $I = f(V)$. Often times, the function describing the I - V relationship is not known beforehand. The function f is assumed to be a polynomial, and the parameters of f (the coefficients for every power of V) are computed using least squares.

Throughout this problem, we are provided with \vec{x} , a set of voltage measurements, and \vec{y} , a set of current measurements.

- Let's first try to model a resistor I - V curve. Run the code in the attached IPython notebook. What is the degree of the polynomial that fits an ideal resistor I - V curve? Play around with degree in the IPython notebook and observe the best fit polynomial's shape. Is the noise influencing the higher degree polynomials?
- The attached IPython notebook provides functions `data_matrix`, `leastSquares` that allow you to fit polynomials of different degrees to the data provided. We also provide a function `cost` that computes the squared error of the fit. In the attached IPython notebook, plot the cost of various degree polynomials fitting to the measured I - V data points for a resistor using the given `cost` function. The `cost` function returns $\|\vec{y} - \mathbf{A}\vec{f}\|^2$, i.e. the squared magnitude of the error vector.

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots \\ 1 & x_2 & x_2^2 & \dots \\ 1 & x_3 & x_3^2 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

As seen above \mathbf{A} is the appropriately sized matrix containing powers of the elements of \vec{x} and the vector \vec{f} contains entries f_n that are the coefficients for the n th power of the elements of \vec{x} . Comment on the shape of the “Cost vs. Degree” graph. Do we want to choose a best fit polynomial of degree greater than one if the cost is lower than the polynomial of degree one? Should we choose the degree of the polynomial based on this graph? This question is meant to make you think, do not worry too much about getting a precise right answer here.

- (c) Now let’s put our hypothesis to the test by extrapolating some data! Using the provided code, graph the least squares polynomial fit IV characteristic for $V = [0,20]$ for both degree 1 and degree 15 polynomials. Which degree fits the extrapolated data better?

7. Image Analysis

Learning Goal: *This problem introduces a method of fitting a non-linear model through a set of measured data points using the least squares method.*

Applications in medical imaging often require an analysis of images based on the image’s pixels. For instance, we might want to count the number of cells in a given biological sample. One way to do this is to take a picture of the cells and use the pixels to determine their locations and how many there are. Automatic detection of shape is useful in image classification as well (e.g. consider a robot trying to find out autonomously where a mug is in its field of vision).

Let us focus back on the medical imaging scenario. You are interested in finding the exact position and shape of a cell in an image. You will do this by finding the **equation of the circle or ellipse** that bounds the cell relative to a given coordinate system in the image. Your collaborator uses edge detection techniques to find **a bunch of points that are approximately along the edge of the cell**. We assume that the origin of the coordinate system is in the center of the image with standard axes (x,y) and your collaborator gives you the following points that approximately bound the cell:

$(0.3, -0.69), (0.5, 0.87), (0.9, -0.86), (1, 0.88), (1.2, -0.82), (1.5, 0.64), (1.8, 0)$.

Recall that an equation of the form

$$a_1x^2 + b_1xy + c_1y^2 + d_1x + e_1y = 1$$

can be used to represent an ellipse (if $b_1^2 - 4a_1c_1 < 0$), and an equation of the form

$$a_1(x^2 + y^2) + d_1x + e_1y = 1$$

is a circle if $d_1^2 + e_1^2 + 4a_1 > 0$. Notice that the circle has fewer parameters.

You don’t need to consider these constraints in your least squares setup, but you are encouraged to check whether your least squares solutions satisfy these constraints.

- (a) How can you find the equation of a *circle* that surrounds the cell by fitting the data points? First, provide a setup and formulate a minimization problem to do this, i.e. a least squares problem minimizing the squared error $\|\mathbf{A}\vec{v} - \vec{b}\|^2$, where you attempt to find the **unknown coefficients** a_1 , d_1 , and e_1 from

your data points. Here your unknown vector $\vec{v} = \begin{bmatrix} a_1 \\ d_1 \\ e_1 \end{bmatrix}$. *Hint: The quantities $(x^2 + y^2)$, x , and y can be thought of as known values calculated from your data points.*

You do not need to simplify the numerical values for the matrix elements; just writing out the matrix with numerical expressions will suffice.

- (b) How can you find the equation of an ellipse (instead of a circle) that surrounds the cell? Provide a setup and formulate a minimization problem similar to that in part (a). Now the unknown vector \vec{v} will be different from the earlier parts.

You do not need to simplify the numerical values for the matrix elements; just writing out the numerical expressions will suffice.

- (c) In the IPython notebook, run the widget and try to trace the outline of the example cell image with "measurements" by clicking the edge of the cell. Then try both fitting with circle and an ellipse. Which shape fits better, and why?
- (d) Now let's try this ourselves with the data given in the problem. In the IPython notebook, write a short program that uses least-squares to fit a circle to the given points. A helper function `plot_circle` is provided. What is $\frac{\|\vec{e}\|}{N}$, where $\vec{e} = \mathbf{A}\vec{v} - \vec{b}$ and N is the number of data points? Plot your points and the best fit circle in IPython.
- (e) In the IPython notebook, write a short program that uses least-squares to fit an ellipse to the given points. A helper function `plot_ellipse` is provided. What is $\frac{\|\vec{e}\|}{N}$, where $\vec{e} = \mathbf{A}\vec{v} - \vec{b}$ and N is the number of data points? Now the unknown vector \vec{v} will be different from the earlier parts. Plot your points and the best fit ellipse in IPython. How does this error compare to the one in the previous subpart?

8. Homework Process and Study Group

Who did you work with on this homework? List names and student ID's. (In case you met people at homework party or in office hours, you can also just describe the group.) How did you work on this homework? If you worked in your study group, explain what role each student played for the meetings this week.