EECS 16A Designing Information Devices and Systems I
Summer 2020 Homework 6B

This homework is due Sunday, August 9, 2020, at 23:59.
Self-grades are due Wednesday August 12, 2020, at 23:59.

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

- `hw6B.pdf`: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook (if any) saved as a PDF.

Homework Learning Goals: *One of the objectives of this homework is to show how correlation can be utilized for positioning. Another aim is to familiarize you with the concepts of projection and least squares. These concepts will be used towards solving overdetermined systems of equations affected by noise.*

1. Mechanical: Projections

(a) Find the projection of $\vec{b} = \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix}$ onto $\vec{a} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$. What is the squared error between the projection and $\vec{b}$, i.e. $\|e\|^2 = \|\text{proj}_{\vec{a}}(\vec{b}) - \vec{b}\|^2$?

Solution:

$$\text{proj}_{\vec{a}}(\vec{b}) = \frac{\langle \vec{b}, \vec{a} \rangle}{\|\vec{a}\|^2} \vec{a} = \frac{\vec{b}^T \vec{a}}{\|\vec{a}\|^2} \vec{a} \tag{1}$$

First, compute $\|\vec{a}\|^2 = \langle \vec{a}, \vec{a} \rangle = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 2.$

Second, compute $\langle \vec{b}, \vec{a} \rangle = \begin{bmatrix} 3 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = 2.$

Plugging in, $\text{proj}_{\vec{a}}(\vec{b}) = \frac{2\vec{a}}{2} = \vec{a}$.

The squared error between $\vec{b}$ and its projection onto $\vec{a}$ is $\|e\|^2 = \|\vec{a} - \vec{b}\|^2 = 12$.

(b) Find the projection of $\vec{b} = \begin{bmatrix} 1 \\ 4 \\ -5 \end{bmatrix}$ onto the subspace defined by the vectors $\left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right\}$. What is the projection's squared error with $\vec{b}$, i.e. $\|e\|^2 = \|\text{proj}_{\vec{a}}(\vec{b}) - \vec{b}\|^2$?

Solution: Let $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\vec{x} \in \mathbb{R}^2$ such that the projection of $\vec{b}$ onto the column space of $\mathbf{A}$ is $\mathbf{A}\vec{x}$.

We will compute $\hat{\vec{x}}$ by solving the following least squares problem,

$$\min_{\vec{x}} \|\mathbf{A}\vec{x} - \vec{b}\|^2 \tag{2}$$

The solution yields,

$$\hat{\vec{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b} \tag{3}$$

$$= \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}^T \begin{bmatrix} 1 \\ 4 \\ -5 \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -4 \\ 4 \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} -2 \\ 4 \end{bmatrix} \tag{6}$$

Plugging in, the projection of $\vec{b}$ onto the column space of $\mathbf{A}$ is $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ 4 \end{bmatrix} = \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix}$.

The squared error between the projection and $\vec{b}$ is $\|\vec{e}\|^2 = \left\| \begin{bmatrix} -2 \\ 4 \\ -2 \end{bmatrix} - \begin{bmatrix} 1 \\ 4 \\ -5 \end{bmatrix} \right\|^2 = 18.$

2. Mechanical Trilateration

Trilateration is the problem of finding one's coordinates given distances from known location coordinates. For each of the following trilateration problems, you are given 3 positions and the corresponding distance from each position to your location. Find your location or possible locations. If a solution does not exist, state that it does not.

(a) $\vec{s}_1 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}, d_1 = 5, \vec{s}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, d_2 = 2\ , \vec{s}_3 = \begin{bmatrix} -11 \\ 6 \end{bmatrix}, d_3 = 13.$

Solution: First the problem will be done in abstract so that we can immediately write the linear system of equations for all three parts. However, if you solved directly using concrete values, give yourself full credit.

$$\|\vec{x} - \vec{s}_1\|^2 = d_1^2$$
$$\|\vec{x} - \vec{s}_2\|^2 = d_2^2$$
$$\|\vec{x} - \vec{s}_3\|^2 = d_3^2$$

We can expand each left hand side out in terms of the definition of the norm:

$$\|\vec{x} - \vec{s}_i\|^2 = \langle \vec{x} - \vec{s}_i, \vec{x} - \vec{s}_i \rangle = (\vec{x} - \vec{s}_i)^T (\vec{x} - \vec{s}_i)$$

$$\vec{x}^T\vec{x} - 2\vec{x}^T\vec{s}_1 + \vec{s}_1^T\vec{s}_1 = d_1^2$$
$$\vec{x}^T\vec{x} - 2\vec{x}^T\vec{s}_2 + \vec{s}_2^T\vec{s}_2 = d_2^2$$
$$\vec{x}^T\vec{x} - 2\vec{x}^T\vec{s}_3 + \vec{s}_3^T\vec{s}_3 = d_3^2$$

Finally, take one equation and subtract it from the other two to get a system of linear equations in $\vec{x}$:

$$2\vec{x}^T\vec{s}_3 - 2\vec{x}^T\vec{s}_1 = d_1^2 - d_3^2 + \vec{s}_3^T\vec{s}_3 - \vec{s}_1^T\vec{s}_1$$
$$2\vec{x}^T\vec{s}_3 - 2\vec{x}^T\vec{s}_2 = d_2^2 - d_3^2 + \vec{s}_3^T\vec{s}_3 - \vec{s}_2^T\vec{s}_2$$

We can express as a matrix equation in $\vec{x}$:

$$\begin{bmatrix} 2(\vec{s}_3 - \vec{s}_1)^T \\ 2(\vec{s}_3 - \vec{s}_2)^T \end{bmatrix} \vec{x} = \begin{bmatrix} d_1^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_1\|^2 \\ d_2^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_2\|^2 \end{bmatrix}$$

We have that:

$$2(\vec{s}_3 - \vec{s}_1) = \begin{bmatrix} -30 \\ 2 \end{bmatrix}$$
$$2(\vec{s}_3 - \vec{s}_2) = \begin{bmatrix} -24 \\ 14 \end{bmatrix}$$
$$d_1^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_1\|^2 = 25 - 169 + 157 - 41 = -28$$
$$d_2^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_2\|^2 = 4 - 169 + 157 - 2 = -10$$

Which gives us the system $\begin{bmatrix} -30 & 2 \\ -24 & 14 \end{bmatrix} \vec{x} = \begin{bmatrix} -28 \\ -10 \end{bmatrix}$ with solution $\vec{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

A solution existing for this system of linear equations does not necessarily guarantee consistency of the system of nonlinear equations, but we can validate:

$$\left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 4 \\ 5 \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} -3 \\ -4 \end{bmatrix} \right\|^2 = 25 = d_1^2$$
$$\left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} 0 \\ 2 \end{bmatrix} \right\|^2 = 4 = d_2^2$$
$$\left\| \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -11 \\ 6 \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} 12 \\ -5 \end{bmatrix} \right\|^2 = 169 = d_3^2$$

(b) $\vec{s}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, d_1 = 5\sqrt{2}, \vec{s}_2 = \begin{bmatrix} 10 \\ 0 \end{bmatrix}, d_2 = 5\sqrt{2}\,, \vec{s}_3 = \begin{bmatrix} 5 \\ 0 \end{bmatrix}, d_3 = 5.$

Solution: Using what was shown in part (a) we have that:

$$2(\vec{s}_3 - \vec{s}_1) = \begin{bmatrix} 10 \\ 0 \end{bmatrix}$$
$$2(\vec{s}_3 - \vec{s}_2) = \begin{bmatrix} -10 \\ 0 \end{bmatrix}$$
$$d_1^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_1\|^2 = 50 - 25 + 25 - 0 = 50$$
$$d_2^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_2\|^2 = 50 - 25 + 25 - 100 = -50$$

Which gives us the system $\begin{bmatrix} 10 & 0 \\ -10 & 0 \end{bmatrix} \vec{x} = \begin{bmatrix} 50 \\ -50 \end{bmatrix}$ with solution $\vec{x} = \begin{bmatrix} 5 \\ \alpha \end{bmatrix}$. However, not all values of α are valid, so we check with the third distance equation:

$$\left\| \begin{bmatrix} 5 \\ \alpha \end{bmatrix} - \begin{bmatrix} 5 \\ 0 \end{bmatrix} \right\|^2 = 5^2 \implies \alpha^2 = 25 \implies \alpha = \pm 5$$

The system of nonlinear equations is consistent with this solution. We do not have enough information to uniquely determine our location, but we know we are at either $\vec{x} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$ or $\vec{x} = \begin{bmatrix} 5 \\ -5 \end{bmatrix}$.

(c) $\vec{s}_1 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}, d_1 = 5, \vec{s}_2 = \begin{bmatrix} 0 \\ -2 \end{bmatrix}, d_2 = 2\ , \vec{s}_3 = \begin{bmatrix} -12 \\ 5 \end{bmatrix}, d_3 = 12.$

Solution: Using again what was shown in part (a) we have that:

$$2(\vec{s}_3 - \vec{s}_1) = \begin{bmatrix} -30 \\ 2 \end{bmatrix}$$
$$2(\vec{s}_3 - \vec{s}_2) = \begin{bmatrix} -24 \\ 14 \end{bmatrix}$$
$$d_1^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_1\|^2 = 25 - 144 + 169 - 25 = 25$$
$$d_2^2 - d_3^2 + \|\vec{s}_3\|^2 - \|\vec{s}_2\|^2 = 4 - 144 + 169 - 4 = 25$$

Which gives us the system $\begin{bmatrix} -30 & 2 \\ -24 & 14 \end{bmatrix} \vec{x} = \begin{bmatrix} 25 \\ 25 \end{bmatrix}$. While a solution, $\vec{x} = \begin{bmatrix} -\frac{75}{93} \\ \frac{75}{186} \end{bmatrix}$, for this system of linear equations exists, it will yield inconsistent distances when substituted back into the nonlinear equations. Therefore there is no solution.

3. Image Analysis

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

$$ax^2 + bxy + cy^2 + dx + ey = 1$$

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

$$a(x^2 + y^2) + dx + ey = 1$$

is a circle if $d^2 + e^2 + 4a > 0$. Notice that the circle has fewer parameters.

(a) How can you find the equation of a *circle* that surrounds the cell? First, provide a setup and formulate a minimization problem to do this, i.e. a least squares problem minimizing the squared error $||\mathbf{A}\vec{x} - \vec{b}||$ where you attempt to find the unknown coefficients a, d, and e from your points. *Hint: The quantities $(x^2 + y^2)$, x, and y can be thought of as variables calculated from your data points.*

Solution:

The setup is:

$$\min_{a,d,e} \left|\left| \begin{bmatrix} x^2+y^2 & x & y \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ d \\ e \end{bmatrix} - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right|\right|$$

We plug in numbers to get:

$$\min_{a,d,e} \left|\left| \begin{bmatrix} 0.5661 & 0.3 & -0.69 \\ 1.0069 & 0.5 & 0.87 \\ 1.5496 & 0.9 & -0.86 \\ 1.7744 & 1 & 0.88 \\ 2.1124 & 1.2 & -0.82 \\ 2.6596 & 1.5 & 0.64 \\ 3.24 & 1.8 & 0 \end{bmatrix} \begin{bmatrix} a \\ d \\ e \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right|\right|$$

(b) How can you find the equation of an ellipse that surrounds the cell? Provide a setup and formulate a minimization problem similar to that in part (a).

Solution:

The setup is:

$$\min_{a,b,c,d,e} \left|\left| \begin{bmatrix} x^2 & xy & y^2 & x & y \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} - \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right|\right|$$

We then plug in values to get:

$$\min_{a,b,c,d,e} \left|\left| \begin{bmatrix} 0.09 & -0.207 & 0.4761 & 0.3 & -0.69 \\ 0.25 & 0.435 & 0.7569 & 0.5 & 0.87 \\ 0.81 & -0.774 & 0.7396 & 0.9 & -0.86 \\ 1 & 0.88 & 0.7744 & 1 & 0.88 \\ 1.44 & -0.984 & 0.6724 & 1.2 & -0.82 \\ 2.25 & 0.96 & 0.4096 & 1.5 & 0.64 \\ 3.24 & 0 & 0 & 1.8 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right|\right|$$

(c) In the IPython notebook, write a short program to fit a circle to the given points. What is $\frac{\|\vec{e}\|}{N}$, where $\vec{e} = \mathbf{A}\vec{x} - \vec{b}$ and N is the number of data points? Plot your points and the best fit circle in IPython.

Solution:

See the IPython notebook.

The solution vector is:

$$\vec{x} = \begin{bmatrix} 4.87 \\ -7.89 \\ -0.23 \end{bmatrix}$$

Thus, we would predict the equation of the circle to be: $4.87(x^2+y^2) - 7.89x - 0.23y = 1$.
This gives the normalized error: $\frac{0.96}{7} = 0.137$.

(d) In the IPython notebook, write a short program to fit an ellipse to the given points. What is $\frac{\|\vec{e}\|}{N}$, where $\vec{e} = \mathbf{A}\vec{x} - \vec{b}$ and N is the number of data points? Plot your points and the best fit ellipse in IPython. How does this error compare to the one in the previous subpart? Which technique is better?

Solution:

See the IPython notebook.

The solution vector is:

$$\vec{x} = \begin{bmatrix} 4.10 \\ 0.49 \\ 4.94 \\ -6.85 \\ -0.62 \end{bmatrix}$$

We predict the general equation to be: $4.10x^2 + 0.49xy + 4.94y^2 - 6.85x - 0.62y = 1$.
This gives the normalized error: $\frac{0.090}{7} = 0.0128$.

The ellipse is a better fit because it has more parameters, so the least squares technique can tune the parameters to be closer to the observations.

4. GPS Receivers

The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. In this problem, we will understand how a receiver (e.g. your cellphone) can disambiguate signals from the different GPS satellites that are simultaneously received.

GPS satellites employ "spread-spectrum" technology (very similar to Code-division multiple access, CDMA, which is commonly used in cellphone transmissions) and a special coding scheme where each transmitter is assigned a code that serves as its "signature".

Each GPS satellite uses a unique 1023 element long sequence as its "signature." These codes used by the satellites are called "Gold codes," and they have some special properties:

- The auto-correlation of a Gold code (correlation with itself) is very **high**.
- The cross-correlation between different Gold codes is very low, i.e. different Gold codes are almost orthogonal to each other.

Gold codes are generated using a linear feedback shift register (LFSR). Understanding how this works is out of scope for the class, but you can read more about LFSR and CDMA if you are interested.

The important thing to know is that the Gold codes are 1023 element vectors where each element is either $+1$ or -1, and that any Gold code is "almost orthogonal" to any other Gold code.

A receiver listening for signature transmissions from a satellite has copies of all of the different GPS satellites' Gold codes. The receiver can determine how long it took for a particular GPS satellite's signal to reach it by taking the correlation of the received signal with a satellite's Gold code. The shift value (delay) that corresponds to maximizing the correlation determines the "propagation delay" between when the GPS satellite transmitted it's signal and when the receiver received it. This time delay can then be converted into a distance (in the case of GPS, electromagnetic waves are used for transmissions, distance is equal to the speed of light multiplied by the time delay).

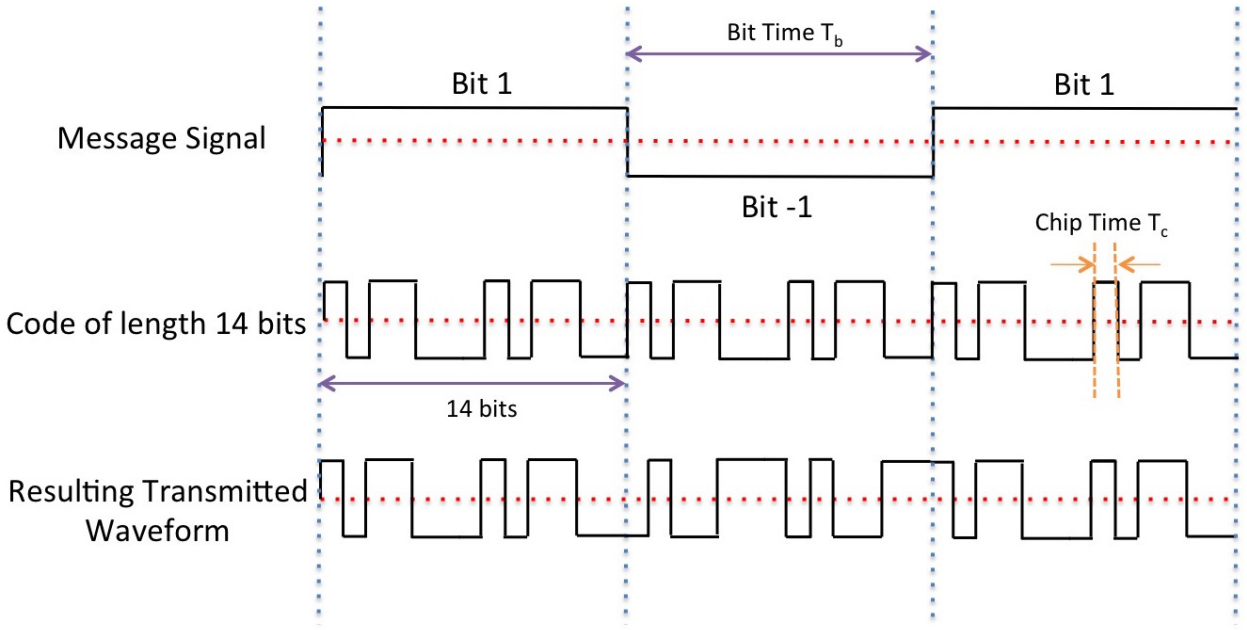
The GPS satellite is constantly transmitting its signature. In addition to identifying itself through its signature, it can also "modulate" the signature to communicate more information. Modulating a signature means multiplying the entire signature block by $+1$ or -1, as shown in the figure.

In the figure below, the signature is of length 14, i.e. it is made of 14 ± 1 symbols. T_c is the duration of one symbol, and T_b is the duration of a whole signature. The figure shows 3 blocks of length 14 being transmitted. The message signal (made of $+1$ and -1 as well) multiplies the entire block of the signature, to give the resulting transformed waveform at the bottom of the figure. The message being transmitted in the figure is $\begin{bmatrix} 1 & -1 & 1 \end{bmatrix}$. So to send these three symbols of message, we need to send 14×3 symbols of the gold code.

The waveform that is actually transmitted is the multiplication of the message signal with the signature signal. Now, when a receiver receives a signal, in addition to finding the time delay between transmission and reception, the receiver will be able to decode the message by noting a very high correlation if the message bit is equal to 1, and a very negative correlation if the message bit is equal to -1.

For the problem you will now do, $T_b = 1023 T_c$. (In reality, $T_b = 20 \times 1023 \times T_c$.)

You will use the ideas of linear correlation to figure out which of the satellites are transmitting.



For the purpose of this question we only consider 24 GPS satellites. Download the IPython notebook and the corresponding data files for the following questions:

Note: this code is calculation-heavy, and can take up to a few mins to run for each code block. Be patient!

(a) Auto-correlate (i.e. cross-correlate with itself) the Gold code of satellite 10 and plot it. Python has functions for this. What do you observe?

Solution: The autocorrelation peaks at 1023 when the signals are perfectly aligned (offset 0). The correlation of a Gold code with a shifted version of itself is not significant.

(b) Cross-correlate the Gold code of satellite 10 with satellite 13 and plot it. What do you observe?

Solution: We see that the cross-correlation of a Gold code of any satellite with any other satellite is very low. This indicates that when given some unknown data, we can differentiate between different satellites.

(c) Consider a random signal, i.e. a signal that is not generated due to a specific code but is a random ± 1 sequence. A helper function in the notebook will generate this for you. Cross-correlate it with the Gold code of satellite 10. What do you observe? What does this mean about our ability to identify satellites in the presence of random ± 1 noise?

Solution: We see that the cross-correlation of the Gold code of any satellite with integer noise is very low. This indicates that we can still figure out the presence of a satellite even if it is buried in noise.

(d) The signal actually received by a receiver will be the satellites' transmissions plus additive noise, and this need not be just noise that takes values ± 1. Use the helper function in the notebook to generate a random noise sequence of length 1023, and compute the cross-correlation of this sequence with the Gold Code of satellite 10. What does this mean about our ability to identify satellites in the presence of real-valued noise?

For the next subparts of this problem, the received signals are corrupted by real-valued noise. Use the observation from this subpart for solving the rest of the question.

Solution: We see that the Gold code of any satellite with Gaussian noise is very low. This indicates that we can still figure out the presence of a satellite even if it is buried in Gaussian noise.

(e) The receiver may receive signals from multiple satellites simultaneously, in which case the signals will all be added together. In addition, noise might be added to the signal. What are the satellites present in `data1.npy`?

Solution: The satellites that are present are satellites 4, 7, 13, and 19.

(f) Let's assume that you can hear only one satellite, Satellite X, at the location you are in (though this never happens in reality). Let's also assume that this satellite is transmitting an unknown sequence of $+1$ and -1 of length 5 (after encoding it with the 1023 bit Gold code corresponding to Satellite X). Find out from `data2.npy` which satellite it is and what sequence of ± 1's it is transmitting.

Solution: Satellite 3 is transmitting 1, -1, -1, -1, 1.

(g) Signals from different transmitters arrive at the receiver with different delays. We use these delays to figure out the distance between the satellite and receiver.

The signals from different satellites are superimposed on each other with different offsets at the start.

What satellites are you able to see in `data3.npy`? Assume that all satellites begin transmission at time 0. What are the delays of all the satellites that are present? Assume you are told that all the satellites have the same message signal given by $\begin{bmatrix}1 & 1 & -1 & -1 & -1\end{bmatrix}$.

Solution: The satellites present in this data are 5 and 20.

The correlation array index where satellite 5's first peak is located is 253. The correlation array index where satellite 20's first peak is located is 506. These would correspond to delays of 253 and 506.

5. Classification: Targeted promotions

This problem describes a situation that online businesses face regularly. They can only observe the current purchases of a customer but would like to understand the general interests of the customer.

The retail store EehEeh Sixteen would like to create an algorithm that can predict a customer's interests based only on the purchases made by the customer. Based on the interests, the store decides to give the customer a coupon (promotion) that is targeted to the right customer's interests. This can be thought of as a classification problem: given the customer's purchase history, we hope to assign the customer to one of finitely many groups, depending on which promotion we deem to be most suitable for them.

Customer interests are described by a vector: $\vec{s_A} = \begin{bmatrix} \text{party-interest score} \\ \text{family-interest score} \\ \text{student-interest score} \\ \text{office-interest score} \end{bmatrix}$

The store would like to infer this vector for each customer.

(a) Assume we have the interests of a customer c in a vector $\vec{x_c} = \begin{bmatrix} c_{\text{party}} \\ c_{\text{family}} \\ c_{\text{student}} \\ c_{\text{office}} \end{bmatrix}$ and a set of promotions $A_1, A_2, \ldots, A_N$, with their attached vectors of scores $\vec{s_{A_1}}, \vec{s_{A_2}}, \ldots, \vec{s_{A_N}}$ (that are also customer interest vectors).

We would like to select the promotion vector that is closest to the customer interest vector, because this is the promotion that the customer would be the most interested in. To perform this selection, we would like to come up with some measure of similarity. Specifically, we want a function that outputs **a higher value if the two vectors are closer to each other**.

The larger the value of the similarity function between $\vec{x_c}$ and $\vec{s_{A_i}}$ the better suited the promotion is for the customer. You have two choices for the similarity measure:

Distance: $sim_1(\vec{x_c}, \vec{s_A}) = \|\vec{x_c} - \vec{s_A}\|$ is a norm and measures the distance between $\vec{x_c}$ and $\vec{s_A}$. Projection: $sim_2(\vec{x_c}, \vec{s_A}) = \left\langle \vec{x_c}, \frac{\vec{s_A}}{\|\vec{s_A}\|} \right\rangle$ is a normalized inner product. Which one is a better similarity measure? Why?

Solution:

Distance: $sim_1(\vec{x_c}, \vec{s_A}) = \|\vec{x_c} - \vec{s_A}\|$ is not good because the farther these vectors are from each other, the higher the score. Which is the opposite of what we wanted. Projection: $sim_2(\vec{x_c}, \vec{s_A}) = \left\langle \vec{x_c}, \frac{\vec{s_A}}{\|\vec{s_A}\|} \right\rangle$ is a better option because the closer the two vectors, the larger the projection will be. Lock on this choice for the rest of the problem.

(b) Unfortunately, the store does not get to observe each customer's interest vector. It only gets to observe the money the customer spends in four categories: food, movies, art, and books. The store needs to use this information to infer the vector $\vec{s_A}$ for each customer.

The EehEeh Sixteen research division conducted some studies that calculated the distribution of spending for people who are purely interested in only one category. For example, a person who is only interested in Party-spending will have the vector $x_c = [1,\ 0,\ 0,\ 0]^T$, and the spending of this person is given in the first line of Table: 1. Similarly, the remaining rows tell you how a person who is just interested in Family, Students, or Offices will spend.

	Spending Category			
Interest Category	**Food**	**Movies**	**Art**	**Books**
Party	40%	33%	22%	5%
Family	70%	10%	10%	10%
Student	20%	10%	15%	55%
Office	5%	2%	20%	73%

Table 1: The distribution of spending of people in each category.

We want to use this data to infer the interest vectors of customers given their spending, assuming that the spending of each customer is a linear combination of the spending of the "pure customers" (those with interest vectors $[1,\ 0,\ 0,\ 0]^T$, $[0,\ 1,\ 0,\ 0]^T$ etc). Suppose a customer spends $T_{\text{food}}\%$ on food, $T_{\text{movies}}\%$ on movies, $T_{\text{art}}\%$ on art, and $T_{\text{books}}\%$ on books.

Use the information in Table 1 to devise a system of linear equations so you can solve for the customer's preferences, x_c.

Solution:

For a given customer, $T_{\text{food}}\%, T_{\text{movies}}\%, T_{\text{art}}\%$ and $T_{\text{books}}\%$ represent the customer's percent spending on food, movies, art and books, respectively.

The system of linear equations that describes the spending above, assuming the spending are observed:

$$\begin{aligned}
0.4c_{\text{party}} + 0.7c_{\text{family}} + 0.2c_{\text{student}} + 0.05c_{\text{office}} &= T_{\text{food}}\% \\
0.33c_{\text{party}} + 0.1c_{\text{family}} + 0.1c_{\text{student}} + 0.02c_{\text{office}} &= T_{\text{movies}}\% \\
0.22c_{\text{party}} + 0.1c_{\text{family}} + 0.15c_{\text{student}} + 0.2c_{\text{office}} &= T_{\text{art}}\% \\
0.05c_{\text{party}} + 0.1c_{\text{family}} + 0.55c_{\text{student}} + 0.73c_{\text{office}} &= T_{\text{books}}\%
\end{aligned}$$

(c) We will combine the results from the previous parts to complete the partially filled out algorithm below. The algorithm takes the raw spending of a customer, $M_{\text{food}}, M_{\text{movies}}, M_{\text{art}}, M_{\text{books}}$, and the promotion scores, $\vec{s_{A_1}}, \vec{s_{A_2}}, \ldots, \vec{s_{A_N}}$, as inputs. The algorithm's output should be the best promotion for that customer.

For this part, use the second similarity metric from part (a). In lines 2 to 5, we first normalize the spending subtotals to get spending percentages.

Algorithm 1 The EehEeh Sixteen promotions algorithm

1: **procedure** PROMOTION($M_{\text{food}}, M_{\text{movies}}, M_{\text{art}}, M_{\text{books}}, \vec{s_{A_1}}, \vec{s_{A_2}}, \ldots, \vec{s_{A_N}}$)
2: $T_{\text{food}}\% = \frac{M_{\text{food}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$
3: $T_{\text{movies}}\% = \frac{M_{\text{movies}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$
4: $T_{\text{art}}\% = \frac{M_{\text{art}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$
5: $T_{\text{books}}\% = \frac{M_{\text{books}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$
6: Set up and solve the system from part b
7: Assign $\vec{x_c} = \begin{bmatrix} c_{\text{party}} \\ c_{\text{family}} \\ c_{\text{student}} \\ c_{\text{office}} \end{bmatrix}$
8: *Pick promotion A using similarity metric from part a.*
9: Print promotion A
10: **end procedure**

How should we pick the promotion A using the similarity metric from part a? Complete the specification of Algorithm 2 by writing down what step 8 should be.

Solution:

Use algorithm 2 for reference. You may want to simplify things by using simpler notation.

(d) Run the algorithm to figure out what promotion we should give to Jane Doe who spent \$6 on food, \$4 on movies, \$1 on art and \$5 on books. Use the values in Table 1 and assume there are 4 promotions, A_1, A_2, A_3, and A_4, with associated score vectors $\vec{s_{A_1}} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$, $\vec{s_{A_2}} = \begin{bmatrix} \frac{2}{3} \\ -\frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{3} \end{bmatrix}$, $\vec{s_{A_3}} = \begin{bmatrix} -\frac{1}{2} \\ -\frac{1}{2} \\ \frac{5}{2} \\ -\frac{1}{2} \end{bmatrix}$ and $\vec{s_{A_4}} = \begin{bmatrix} 0 \\ \frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}$.

You may use IPython to run your algorithm. *Hint: Note that the preference vectors do not all have the same magnitude!*

Algorithm 2 The EehEeh Sixteen promotions algorithm

procedure PROMOTION($M_{\text{food}}, M_{\text{movies}}, M_{\text{art}}, M_{\text{books}}, \vec{s_{A_1}}, \vec{s_{A_2}}, \ldots, \vec{s_{A_N}}$)

$T_{\text{food}}\% = \frac{M_{\text{food}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$

$T_{\text{movies}}\% = \frac{M_{\text{movies}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$

$T_{\text{art}}\% = \frac{M_{\text{art}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$

$T_{\text{books}}\% = \frac{M_{\text{books}}}{M_{\text{food}}+M_{\text{movies}}+M_{\text{art}}+M_{\text{books}}}$

Solve the system

$$\begin{aligned}
0.4c_{\text{party}} + 0.7c_{\text{family}} + 0.2c_{\text{student}} + 0.05c_{\text{office}} &= T_{\text{food}}\% \\
0.33c_{\text{party}} + 0.1c_{\text{family}} + 0.1c_{\text{student}} + 0.02c_{\text{office}} &= T_{\text{movies}}\% \\
0.22c_{\text{party}} + 0.1c_{\text{family}} + 0.15c_{\text{student}} + 0.2c_{\text{office}} &= T_{\text{art}}\% \\
0.05c_{\text{party}} + 0.1c_{\text{family}} + 0.55c_{\text{student}} + 0.73c_{\text{office}} &= T_{\text{books}}\%
\end{aligned}$$

Assign $\vec{x_c} = \begin{bmatrix} c_{\text{party}} \\ c_{\text{family}} \\ c_{\text{student}} \\ c_{\text{office}} \end{bmatrix}$

Pick promotion A such that $\left\langle \vec{x_c}, \frac{\vec{s_A}}{\|\vec{s_A}\|} \right\rangle$ *is highest.*

Print promotion A

end procedure

Solution:

First, we normalize to get: $T_{\text{food}}\% = 37.5\%, T_{\text{movies}}\% = 25\%, T_{\text{art}}\% = 6.25\%, T_{\text{books}}\% = 31.25\%$ which yields the following system of linear equations:

$$\begin{aligned}
0.4c_{\text{party}} + 0.7c_{\text{family}} + 0.2c_{\text{student}} + 0.05c_{\text{office}} &= 0.375 \\
0.33c_{\text{party}} + 0.1c_{\text{family}} + 0.1c_{\text{student}} + 0.02c_{\text{office}} &= 0.25 \\
0.22c_{\text{party}} + 0.1c_{\text{family}} + 0.15c_{\text{student}} + 0.2c_{\text{office}} &= 0.0625 \\
0.05c_{\text{party}} + 0.1c_{\text{family}} + 0.55c_{\text{student}} + 0.73c_{\text{office}} &= 0.3125
\end{aligned}$$

The solution to this system is $\vec{x_c} = \begin{bmatrix} -0.023 \\ -0.223 \\ 3.187 \\ -1.941 \end{bmatrix}$

Running sim_2 we get $\left\langle \vec{x_c}, \frac{\vec{s_{A_1}}}{\|\vec{s_{A_1}}\|} \right\rangle = -2.687, \left\langle \vec{x_c}, \frac{\vec{s_{A_2}}}{\|\vec{s_{A_2}}\|} \right\rangle = 1.015, \left\langle \vec{x_c}, \frac{\vec{s_{A_3}}}{\|\vec{s_{A_3}}\|} \right\rangle = 3.425$ and $\left\langle \vec{x_c}, \frac{\vec{s_{A_4}}}{\|\vec{s_{A_4}}\|} \right\rangle = -1.530$ and therefore, the promotion A_3 will be printed.

(e) Will there ever be a customer for which the system devised in part (b) will yield no solutions or infinite solutions?

Solution:

No. Let us work with what we know about invertibility of a matrix. Specifically, we know that a matrix is invertible when the columns of that matrix are linearly independent. In this problem, the vectors on the columns of our system matrix represent the spending percentages for a person "purely" interested in a single topic. This implies each "pure" person's spending vectors will not be collinear with anyone

elses' spending vector (otherwise those people would be interested in the same topic). In our course's terminology, this implies that the columns are linearly independent, that the matrix is invertible, and their will be single unique solution for each customer.

6. Homework Process and Study Group

Who else did you work with on this homework? List names and student ID's. (In case of homework party, you can also just describe the group.) How did you work on this homework?

Solution:

I worked on this homework with...

I first worked by myself for 2 hours, but got stuck on problem 5, so I went to office hours on...

Then I went to homework party for a few hours, where I finished the homework.