EECS 16A Designing Information Devices and Systems I
Fall 2017 Homework 2

This homework is due September 11, 2017, at 23:59.
Self-grades are due September 14, 2017, at 23:59.

Submission Format

Your homework submission should consist of **two** files.

- `hw2.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 of your IPython notebook, you will not receive credit for problems that involve coding. Make sure that your results and your plots are visible.

- `hw2.ipynb`: A single IPython notebook with all of your code in it.

  In order to receive credit for your IPython notebook, you must submit both a "printout" and the code itself.

Submit each file to its respective assignment on Gradescope.

1. (PRACTICE) Powers Of Nilpotent Matrices

Do this problem if you would like more mechanical practice with matrix multiplication.
The following matrices are examples of a special type of matrix called a nilpotent matrix. What happens to each of these matrices when you multiply it by itself four times? Multiply them to find out. Why do you think these are called "nilpotent" matrices? (Of course, there is nothing magical about 4×4 matrices. You can have nilpotent square matrices of any dimension greater than 1.)

(a) Calculate $\mathbf{A}^4$ by hand. Make sure you show what $\mathbf{A}^2$ and $\mathbf{A}^3$ are along the way.

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Calculate $\mathbf{B}^4$ by hand. Make sure you show what $\mathbf{B}^2$ and $\mathbf{B}^3$ are along the way.

$$\mathbf{B} = \begin{bmatrix} 3 & 4 & 2 & 1 \\ -5 & -6 & -3 & -1 \\ 6 & 7 & 3 & 2 \\ 2 & 2 & 1 & 0 \end{bmatrix}$$

2. Elementary Matrices

This week, we learned about an important technique for solving systems of linear equations called Gaussian elimination. It turns out that each row operation in Gaussian elimination can be performed by multiplying the augmented matrix on the left by a specific matrix called an *elementary matrix*. For example, suppose we want to row reduce the following augmented matrix:

$$\mathbf{A} = \left[\begin{array}{cccc|c} 1 & -2 & 0 & -5 & 16 \\ 0 & 1 & 0 & 3 & -7 \\ -2 & -3 & 1 & -6 & 9 \\ 0 & 1 & 0 & 2 & -5 \end{array}\right] \tag{1}$$

What matrix do you get when you subtract the 4th row from the 2nd row of $\mathbf{A}$ (putting the result in row 2)? (You don't have to include this in your solution.) Now, try multiplying the original $\mathbf{A}$ on the left by

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(You don't have to include this in your solutions either.) Notice that you get the same thing.

$$\mathbf{EA} = \left[\begin{array}{cccc|c} 1 & -2 & 0 & -5 & 16 \\ 0 & 0 & 0 & 1 & -2 \\ -2 & -3 & 1 & -6 & 9 \\ 0 & 1 & 0 & 2 & -5 \end{array}\right]$$

$\mathbf{E}$ is a special type of matrix called an *elementary matrix*. This means that we can obtain the matrix $\mathbf{E}$ from the identity matrix by applying an elementary row operation – in this case, subtracting the 4th row from the 2nd row.

In general, any elementary row operation can be performed by left multiplying by an appropriate elementary matrix. In other words, you can perform a row operation on a matrix $\mathbf{A}$ by first performing that row operation on the identity matrix to get an elementary matrix, and then left multiplying $\mathbf{A}$ by the elementary matrix (like we did above).

(a) Write down the elementary matrices required to perform the following row operations on a 4×5 augmented matrix.

- Switching rows 1 and 3
- Multiplying row 3 by -5
- Adding $3\times$row 2 to row 4 (putting the result in row 4) and subtracting row 2 from row 1 (putting the result in row 1)

*Hint: For the last one, note that if you want to perform two row operations on the matrix **A**, you can perform them both on the identity matrix and then left multiply **A** by the resulting matrix.*

(b) Now, compute a matrix $\mathbf{E}$ (by hand) that fully row reduces the augmented matrix $\mathbf{A}$ given in Equation 1 – that is, find $\mathbf{E}$ such that $\mathbf{EA}$ is in reduced row echelon form. Show that this is true by multiplying out $\mathbf{EA}$. When an augmented matrix is in reduced row echelon form, it will have the form

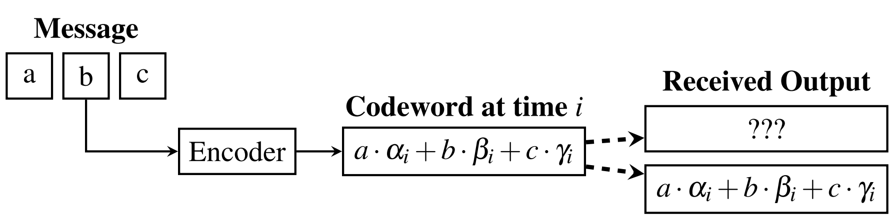
$$\mathbf{EA} = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & b_1 \\ 0 & 1 & 0 & 0 & b_2 \\ 0 & 0 & 1 & 0 & b_3 \\ 0 & 0 & 0 & 1 & b_4 \end{array}\right]$$

*Hint: As before, note that you can either **apply a set of row operations to the same identity matrix** or **apply them to separate identity matrices and then multiply the matrices together**. Make sure, though, that you apply the row operations and multiply the matrices in the correct order.*

3. Fountain Codes

Consider a sender, Alice, and a receiver, Bob. Alice wants to send a message to Bob, but the message is too big to send all at once. Instead, she breaks her message up into little chunks that she sends across a wireless channel one at a time (think radio transmitter to antenna). She knows some of the packets will be corrupted or erased along the way (someone might turn a microwave on...), so she needs a way to protect her message from errors. This way, even if Bob gets a message missing parts of words, he can still figure out what Alice is trying to say! One coding strategy is to use fountain codes. Fountain codes are a type of error-correcting codes based on principles of linear algebra. They were actually developed right here at Berkeley! The company that commercialized them, Digital Fountain, (started by a **Berkeley grad, Mike Luby**), was later acquired by Qualcomm. In this problem, we will explore some of the underlying principles that make fountain codes work in a very simplified setting.

In this problem, we concentrate on the case with transmission erasures, i.e. where a bad transmission causes some parts the message to be erased. Let us say Alice wants to convey the set of her three favorite ideas covered in EE16A lecture each day to Bob. For this, she maps each idea to a real number and wants to convey the 3-tuple $\begin{bmatrix} a & b & c \end{bmatrix}^T$ (Let us say there are an infinite number of ideas covered in EE16A). At each time step, she can send one number, which we will call a "symbol" across, so one possible way for her to send the message is to first send a, then send b, and then send c. However, this method is particularly susceptible to losses. For instance, if the first symbol is lost, then Bob will receive $\begin{bmatrix} ? & b & c \end{bmatrix}^T$, and he will have no way of knowing what Alice's favorite idea is.



(a) The main idea in coding for erasures is to send redundant information, so that we can recover from losses. Thus, if we have three symbols of information, we might transmit six symbols for redundancy. One of the most naive codes is called the repetition code. Here, Alice would transmit $\begin{bmatrix} a & b & c & a & b & c \end{bmatrix}^T$. How much erasure can this transmission recover from? Are there specific patterns it cannot handle?

(b) A better strategy for transmission is to send linear combinations of symbols. Alice and Bob decide in advance on a collection of vectors $\vec{v}_i^T = \begin{bmatrix} \alpha_i & \beta_i & \gamma_i \end{bmatrix}$, $1 \le i \le 6$. These vectors define the code: at time i, Alice transmits the scalar

$$k_i = \vec{v}_i^T \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \alpha_i a + \beta_i b + \gamma_i c.$$

Formulate the setup/the six transmitted symbols using matrix/vector notation.

(c) What are the vectors $\vec{v}_i^T = \begin{bmatrix} \alpha_i & \beta_i & \gamma_i \end{bmatrix}$, $1 \le i \le 6$ that generate the repetition code strategy in part (a)?

(d) Suppose now they choose a collection of seven vectors:

$$\vec{v}_1^T = \begin{bmatrix}1 & 0 & 0\end{bmatrix}, \quad \vec{v}_2^T = \begin{bmatrix}0 & 1 & 0\end{bmatrix}, \quad \vec{v}_3^T = \begin{bmatrix}0 & 0 & 1\end{bmatrix}, \quad \vec{v}_4^T = \begin{bmatrix}1 & 1 & 0\end{bmatrix},$$
$$\vec{v}_5^T = \begin{bmatrix}1 & 0 & 1\end{bmatrix}, \quad \vec{v}_6^T = \begin{bmatrix}0 & 1 & 1\end{bmatrix}, \quad \vec{v}_7^T = \begin{bmatrix}1 & 1 & 1\end{bmatrix}$$

Again, at time i, Alice transmits the scalar $k_i = \vec{v}_i^T \begin{bmatrix}a \\ b \\ c\end{bmatrix}$. Under what conditions, (i.e. what patterns of losses) can Bob still recover the message?

(e) Suppose, using the collection of vectors in part (d), Bob receives $\begin{bmatrix}7 & ? & ? & 3 & 4 & ? & ?\end{bmatrix}^T$. What was the transmitted message? Express the problem as a system of linear equations using matrix/vector notation.

(f) Fountain codes build on these principles. The basic idea used by these codes is that Alice keeps sending linear combinations of symbols until Bob has received enough to decode the message. So at time 1, Alice sends the linear combination using $\vec{v}_1^T$, at time 2 she sends the linear combination using $\vec{v}_2^T$ and so on. After each new linear combination is sent, Bob will send back an acknowledgement *if* he can decode her message (i.e. figure out the original $\begin{bmatrix}a & b & c\end{bmatrix}^T$ that she intended to communicate). So clearly, the minimum number of transmissions for Alice is 3. If Bob receives the first three linear combinations that are sent, Alice is done in three steps! But because of erasures, she might hear nothing (he hasn't decoded yet). Suppose Alice used

$$\vec{v}_1^T = \begin{bmatrix}1 & 0 & 0\end{bmatrix}, \quad \vec{v}_2^T = \begin{bmatrix}0 & 1 & 0\end{bmatrix}, \quad \vec{v}_3^T = \begin{bmatrix}0 & 0 & 1\end{bmatrix}$$

as her first three vectors, but she has still not received an acknowledgement from Bob. Should she choose new vectors according to the strategy in part (a) or the strategy in part (d)? Why?

4. Show It

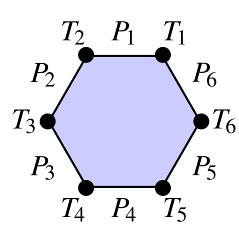
Let n be a positive integer. Let $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k\}$ be a set of k linearly dependent vectors in $\mathbb{R}^n$. Show that for any $n \times n$ matrix $\mathbf{A}$, the set $\{\mathbf{A}\vec{v}_1, \mathbf{A}\vec{v}_2, \ldots, \mathbf{A}\vec{v}_k\}$ is a set of linearly dependent vectors. Make sure that you prove this rigorously for all possible matrices $\mathbf{A}$.

5. Figuring Out The Tips

A number of people gather around a round table for a dinner. Between every adjacent pair of people, there is a plate for tips. When everyone has finished eating, each person places half their tip in the plate to their left and half in the plate to their right. In the end, of the tips in each plate, some of it is contributed by the person to its right, and the rest is contributed by the person to its left. Suppose you can only see the plates of tips after everyone has left. Can you deduce everyone's individual tip amounts?

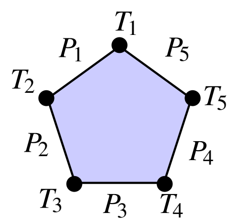
Note: For this question, if we assume that tips are positive, we need to introduce additional constraints enforcing that, and we wouldn't get a linear system of equations. Therefore, we are going to ignore this constraint and assume that negative tips are acceptable.

(a) Suppose 6 people sit around a table and there are 6 plates of tips at the end.



If we know the amounts in every plate of tips (P_1 to P_6), can we determine the individual tips of all 6 people (T_1 to T_6)? If yes, explain why. If not, give two different assignments of T_1 to T_6 that will result in the same P_1 to P_6.

(b) The same question as above, but what if we have 5 people sitting around a table?



(c) If n is the total number of people sitting around a table, for which n can you figure out everyone's tip? You do not have to rigorously prove your answer.

6. Image Stitching

Often, when people take pictures of a large object, they are constrained by the field of vision of the camera. This means that they have two options how they can capture the entire object:

- Stand as far as away as they need to to include the entire object in the camera's field of view (clearly, we do not want to do this as it reduces the amount of detail in the image)
- (This is more exciting) Take several pictures of different parts of the object, and stitch them together, like a jigsaw puzzle.

We are going to explore the second option in this problem. Daniel, who is a professional photographer, wants to construct an image by using "image stitching". Unfortunately, Daniel took some of the pictures from different angles as well as from different positions and distances from the object. While processing these pictures, Daniel lost information about the positions and orientations from which the pictures were taken. Luckily, you and your friend Marcela, with your wealth of newly acquired knowledge about vectors and rotation matrices, can help him!

You and Marcela are designing an iPhone app that stitches photographs together into one larger image. Marcela has already written an algorithm that finds common points in overlapping images and it's your job to figure out how to stitch the images together. You recently learned about vectors and rotation matrices in EE16A, and you have an idea about how to do this.

Your idea is that you should be able to find a single rotation matrix, $\mathbf{R}$, which is a function of some angle, θ, and a translation vector, $\vec{T}$, that transforms every common point in one image to that same point in the other image. Once you find the the angle, θ, and the translation vector, $\vec{T}$, you will be able to transform one image so that it lines up with the other image.

Suppose $\vec{p}$ is a point in one image and $\vec{q}$ is the corresponding point (i.e., they represent the same thing in the scene) in the other image. You write down the following relationship between $\vec{p}$ and $\vec{q}$.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{R}(\theta)} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

This looks good, but then you realize that one of the pictures might be farther away than the other. You realize that you need to add a scaling factor, $\lambda > 0$.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \lambda \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{2}$$

(For example, if $\lambda > 1$, then the image containing q is closer (appears larger) than the image containing p. If $0 < \lambda < 1$, then the image containing q appears smaller.)

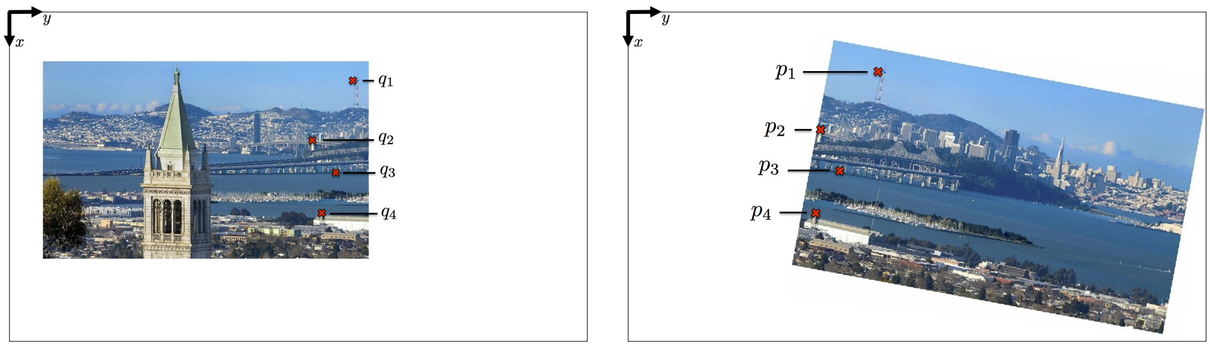


Figure 1: Two images to be stitched together with pairs of matching points labeled.

You are now confident that if you can find θ, $\vec{T}$, and λ, you will be able to reorient and scale one of the images, so that it lines up with the other image.

Before you get too excited, however, you realize that you have a problem. Equation 2 is not a linear equation with respect to θ, $\vec{T}$, and λ. You're worried that you don't have a good technique for solving nonlinear systems of equations. You decide to talk to Marcela and the two of you come up with a brilliant solution.

You decide to "relax" the problem, so that you're solving for a general matrix $\mathbf{R}$ rather than a perfect scaled rotation matrix. The new equation you come up with is:

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} R_{xx} & R_{xy} \\ R_{yx} & R_{yy} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{3}$$

This equation is linear, so you can solve for $R_{xx}, R_{xy}, R_{yx}, R_{yy}, T_x$, and T_y. Also you realize that if $\vec{p}$ and $\vec{q}$ actually do differ by a rotation of θ degrees and a scaling of λ, you can expect that the general matrix $\mathbf{R}$ that you find will turn out to be a scaled rotation matrix with $R_{xx} = \lambda\cos(\theta)$, $R_{xy} = -\lambda\sin(\theta)$, $R_{yx} = \lambda\sin(\theta)$, and $R_{yy} = \lambda\cos(\theta)$.

(a) Multiply Equation 3 out into two scalar linear equations. What are the known values and what are the unknowns in each equation? How many unknowns are there? How many equations do you need to solve for all the unknowns? How many pairs of common points $\vec{p}$ and $\vec{q}$ will you need in order to write down a system of equations that you can use to solve for the unknowns?

(b) Write out a system of linear equations that you can use to solve for the values of $\mathbf{R}$ and $\vec{T}$.

(c) In the IPython notebook `prob2.ipynb`, you will have a chance to test out your solution. Plug in the values that you are given for p_x, p_y, q_x, and q_y for each pair of points into your system of equations to solve for the parameters $\mathbf{R}$ and $\vec{T}$. You will be prompted to enter your results, and the notebook will then apply your transformation to the second image and show you if your stitching algorithm works.

(d) We will now explore when this algorithm fails. For example, the three pairs of points must all be distinct points. Show that if $\vec{p}_1, \vec{p}_2, \vec{p}_3$ are *collinear*, the system of equations (3) is underdetermined. Does this make sense geometrically?
(Think about the kinds of transformations possible by a general affine transformation. An affine transformation is one that preserves points. For example, in the rotation of a line, the angle of the line might change, but the length will not. All linear transformations are affine. **Definition of Affine.**)
Use the following fact: $\vec{p}_1, \vec{p}_2, \vec{p}_3$ are collinear iff $(\vec{p}_2 - \vec{p}_1) = k(\vec{p}_3 - \vec{p}_1)$ for some $k \in \mathbb{R}$.

(e) Show that if the three points are not collinear, the system is fully determined.

(f) Marcela comments that perhaps the system (with three collinear points) is only underdetermined because we "relaxed" our model too much by allowing for general affine transformations, instead of just isotropic-scale/rotation/translation. Can you come up with a different representation of Equation 2, that will allow for recovering the transform from only *two* pairs of distinct points?
(*Hint: Let* $a = \lambda\cos(\theta)$ *and* $b = \lambda\sin(\theta)$. *In other words, enforce* $R_{xx} = R_{yy}$ *and* $R_{xy} = -R_{yx}$).

7. Write Your Own Question And Provide a Thorough Solution.

Writing your own problems is a very important way to really learn material. The famous "Bloom's Taxonomy" that lists the levels of learning is: Remember, Understand, Apply, Analyze, Evaluate, and Create. Using what you know to create is the top level. We rarely ask you any homework questions about the lowest level of straight-up remembering, expecting you to be able to do that yourself (e.g. making flashcards). But we don't want the same to be true about the highest level. As a practical matter, having some practice at trying to create problems helps you study for exams much better than simply counting on solving existing practice problems. This is because thinking about how to create an interesting problem forces you to really look at the material from the perspective of those who are going to create the exams. Besides, this is fun. If you want to make a boring problem, go ahead. That is your prerogative. But it is more fun to really engage with the material, discover something interesting, and then come up with a problem that walks others down a journey that lets them share your discovery. You don't have to achieve this every week. But unless you try every week, it probably won't ever happen.

8. 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?