EECS 16A Designing Information Devices and Systems I
Spring 2017 Babak Ayazifar, Vladimir Stojanovic Homework 2

This homework is due February 6, 2017, at 23:59.
Self-grades are due February 9, 2017, at 23:59.

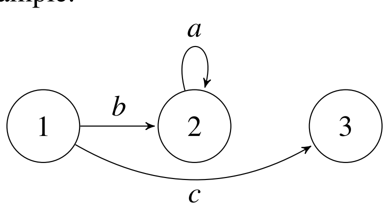
Submission Format
Your homework submission should consist of **two** files.

- `hw2.pdf`: A single pdf file that contains all your answers (any handwritten answers should be scanned) as well as your IPython notebook saved as a pdf. You can do this by printing the IPython notebook page in your browser and selecting the save to pdf option. Make sure any plots and results are showing. Also make sure you combine any separate pdfs into one file.
- `hw2.ipynb`: A single IPython notebook with all your code in it.

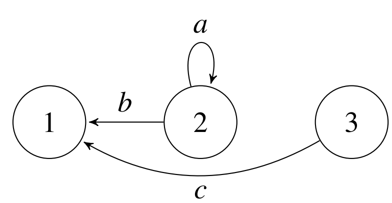
Submit each file to its respective assignment in Gradescope.

1. Transition Matrix Proofs

(a) Suppose there exists some network of websites, such as the "Original" example below. Assume the state vector at some time n is known. Would reversing the arrow directions, as shown in the "Reversed" example below, allow you to find the state vector at time $n-1$? If yes, argue why. If no, provide a counterexample.



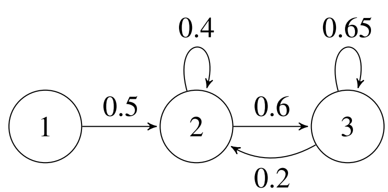
Original



Reverse

(b) Suppose there is a state transition matrix such that the entries of each column vector sum to one. What is the physical interpretation about the total amount of people in the system?

(c) Set up the state transition matrix $\mathbf{A}$ for the network shown below. Explain what this $\mathbf{A}$ matrix implies physically about the total amount of people in this system. (Note: If there is no “self arrow / self loop,” then the people do not return to the original website.)



(d) There is a state transition matrix where the entries of its rows sum to one. Prove that applying this system to a uniform vector will return the same uniform vector. A uniform vector is a vector where all the elements are the same.

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x \\ \vdots \\ x \end{bmatrix} = \begin{bmatrix} x \\ \vdots \\ x \end{bmatrix}$$

2. Show It

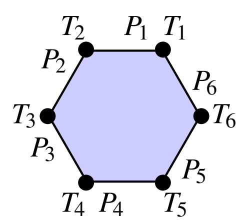
Let n be a positive integer. Let $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k\}$ be a set of k linearly dependent vectors in $\mathbb{R}^n$. Show that for any $n \times n$ matrix $\mathbf{A}$, the set $\{\mathbf{A}\vec{v}_1, \mathbf{A}\vec{v}_2, \ldots, \mathbf{A}\vec{v}_k\}$ is a set of linearly dependent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

3. 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?

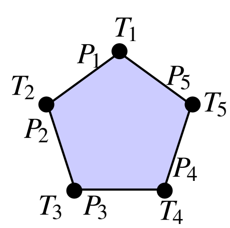
Note: For this question, if we assume that tips are positive, we need to introduce additional constraints enforcing that, and we wouldn't get a linear system of equations. So we are going to ignore this constraint and negative tips are ok.

(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.

4. Image Stitching

Often, when people take pictures of a large object, they are constrained by the field of view of the camera. This means that they have two options by which 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. Prof. Ayazifar, who is a professional photographer, wants to construct an image by doing this "image stitching". Unfortunately, he took some of the pictures at different angles, as well as at different positions and distances from the object. While processing these pictures, he lost information about the positions and orientations at which he took them. 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, **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 $\vec{q}$.

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{\mathbf{R}(\theta)} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \end{bmatrix} \tag{1}$$

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.)

You are now confident that if you can find θ, $\vec{T}$, and λ, then 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 (**??**) is not a linear equation in 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 **R** rather than precisely a 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, 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 **R** that

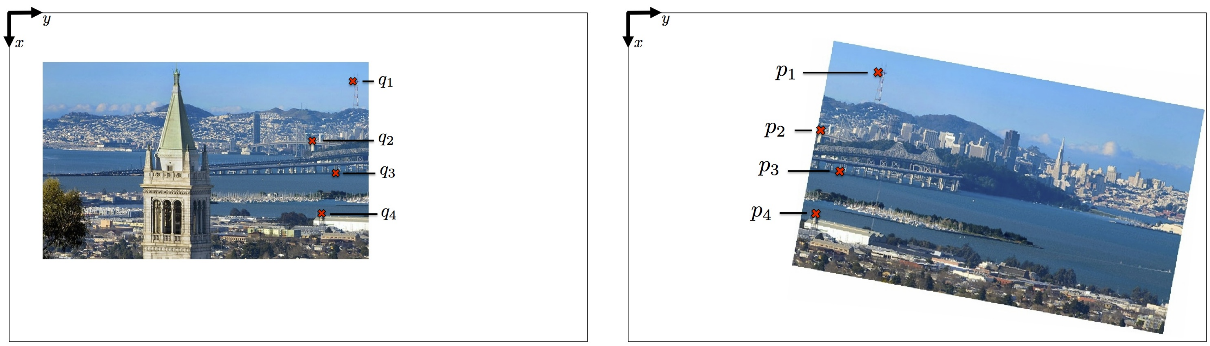


Figure 1: Two images to be stitched together with pairs of matching points labeled.

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 out Equation (**??**) 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 *co-linear*, the system of (**??**) is underdetermined. Does this make sense geometrically?
(Think about the kinds of transformations possible by a general affine transform. An affine transform is one that preserves points, for example in the rotation of a line although the angle of the line might change the length will not. All linear transformations are affine. **Click me! Definition of Affine.**)
Use the fact that: $\vec{p}_1, \vec{p}_2, \vec{p}_3$ are co-linear iff $(\vec{p}_2 - \vec{p}_1) = k(\vec{p}_3 - \vec{p}_1)$ for some $k \in \mathbb{R}$.

(e) *(PRACTICE)* Show that if the three points are not co-linear, the system is fully determined.

(f) *(PRACTICE)* Marcela comments that perhaps the system (with three co-linear points) is only underdetermined because we "relaxed" our model too much by allowing for general affine transforms, instead of just isotropic-scale/rotation/translation. Can you come up with a different representation of (**??**), 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 $R_{xy} = -R_{yx}$).

5. Homework process and study group

Who else did you work with on this homework? List names and student ID's. (In case of hw party, you can also just describe the group.) How did you work on this homework?
Working in groups of 3-5 will earn credit for your participation grade.

6. (PRACTICE) Powers of a Nilpotent Matrix

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

$$\mathbf{A} = \begin{bmatrix} 0 & 2 & 1 & 3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

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

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

7. (PRACTICE) Elementary Matrices

Last week, we learned about an important technique for solving systems of linear equations called Gaussian Elimination. It turns out 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:

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & 0 & -5 & \vdots & 16 \\ 0 & 1 & 0 & 3 & \vdots & -7 \\ -2 & -3 & 1 & -6 & \vdots & 9 \\ 0 & 1 & 0 & 2 & \vdots & -5 \end{bmatrix} \tag{6}$$

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 solutions.) Now, try multiplying the original $\mathbf{A}$ on the left by

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

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

$$\mathbf{EA} = \begin{bmatrix} 1 & -2 & 0 & -5 & \vdots & 16 \\ 0 & 0 & 0 & 1 & \vdots & -2 \\ -2 & -3 & 1 & -6 & \vdots & 9 \\ 0 & 1 & 0 & 2 & \vdots & -5 \end{bmatrix} \tag{8}$$

E is a special type of matrix called an *elementary matrix*. This means that we can obtain the matrix **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 **A** by first performing that row operation on the identity matrix to get an elementary matrix, and then left multiplying **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 2
- Multiplying row 3 by -4
- Adding $2\times$ 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 this last problem, 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 **E** (by hand) that fully row reduces the augmented matrix **A** given in Eqn (**??**) - that is find **E** such that **EA** is in full row reduced echelon form. Show that this is true by multiplying out **EA**. As a reminder in this case, when the augmented matrix is fully row-reduced it will have the form

$$\mathbf{EA} = \begin{bmatrix} 1 & 0 & 0 & 0 & \vdots & b_1 \\ 0 & 1 & 0 & 0 & \vdots & b_2 \\ 0 & 0 & 1 & 0 & \vdots & b_3 \\ 0 & 0 & 0 & 1 & \vdots & b_4 \end{bmatrix} \tag{9}$$

*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 both apply the row operations in the correct order and multiply the matrices in the correct order.*