University of California
College of Engineering
Department of Electrical Engineering
and Computer Sciences

E. Alon, G. Ranade, B. Ayazifar,
C. Tomlin, V. Subramanian

Thurs., Feb. 19, 2015
2:00-3:30pm

EECS 16A: SPRING 2015—MIDTERM 1

Important notes: Please read every question carefully and completely – the setup may or may not be the same as what you have seen before. Also, be sure to show your work since that is the only way we can potentially give you partial credit.

NAME	Last Solutions First

SID	

Problem 1: _____/ 12

Problem 2: _____/ 14

Problem 3: _____/ 13

Problem 4: _____/ 10

Problem 5: _____/ 16

Total: _____/ 65

PROBLEM 1. Imaging Circuit (12 points)

In this problem we will return one final time to examining how one might choose the area of a photodetector in order to maximize the measured voltage. This time however, let's assume that light shines only over a fixed area $A_{illuminated}$.

(Recall that in the rain analogy, the area of the photodetector is analogous to the area of the bucket, the current is analogous to the amount of water per unit time being captured in the bucket, and the voltage is analogous to the height of the water accumulated within the bucket.)

a) **(3 pts)** Let's assume that the area of the photodetector is A_{PD}, that A_{PD} is smaller than $A_{illuminated}$, and that the photodetector is placed such that it fits entirely within the illuminated region. In this case, can you predict how the current produced by the photodetector I_{PD} would change as a function of A_{PD} and $A_{illuminated}$?

In this case increasing A_{PD} captures more light (photons), so:

$$\boxed{I_{PD} \propto \frac{A_{PD}}{A_{illuminated}}}$$

(Note that an answer with $I_{pd} \propto A_{pd}$ will receive full credit as well)

b) **(3 pts)** Now let's assume that A_{PD} is larger than $A_{illuminated}$ and that the illuminated area is fully captured by the photodetector; how does the current I_{PD} depend upon on A_{PD} now?

Now increasing A_{PD} does nothing to change the amount of light that is captured, so:

$$\boxed{I_{PD} \sim \text{Constant}}$$

c) **(6 pts)** Let's now assume that in order to read out its voltage, we have to connect the photodetector device to a circuit with a fixed capacitance, and that we operate the photodetector the same way as we did in homework 0. In other words, before each measurement, we discharge both the photodetector and the measurement circuit (note that you can think of the capacitance of the readout circuit as being the measurement bucket from homework 0; remember that the measurement bucket is not exposed directly to the rain), let the photodetector charge up over a period of time, and then connect it to the readout circuit.

Given your answers to part a) and b), what should you set A_{PD} equal to in order to maximize the measured voltage? Be sure to explain your answer; some equations and/or plots showing how the voltage depends on A_{PD} and $A_{illuminated}$ may be useful.

Just like in the homework, some of the charge collected by the photodetector has to go in to the measurement circuit's capacitance. Conceptually, if that circuit had an equivalent area (remember capacitance $\propto$ area) of A_{meas}, then for $A_{PD} < A_{illuminated}$:

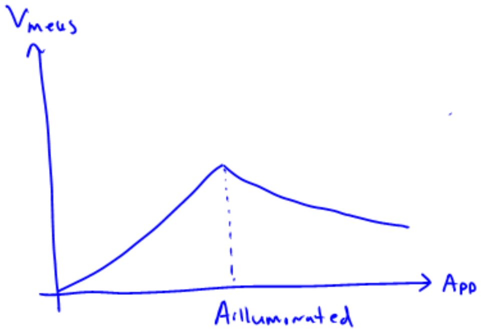
$$V_{meas} \propto \frac{1}{A_{PD}+A_{meas}} \cdot \frac{A_{PD}}{A_{illuminated}}$$

However, increasing A_{PD} above $A_{illuminated}$ doesn't give us any more charge (since I_{PD} is now fixed). So if $A_{PD} > A_{illuminated}$:

$$V_{meas} \propto \frac{A_{illuminated}}{A_{PD}+A_{meas}}$$

Putting these together, the point at which we get maximum V_{meas} is for $\boxed{A_{PD} = A_{illuminated}}$; this choice gives us the maximum charge we can get with the smallest total capacitance.

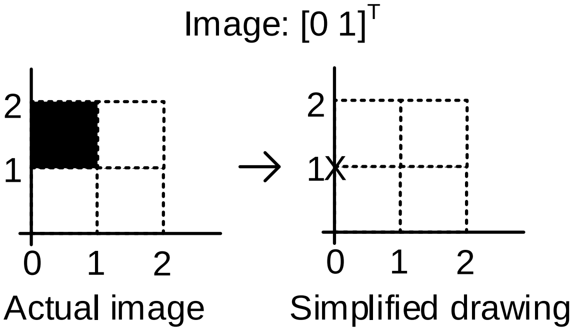
For completeness, if you plot V_{meas} vs. A_{PD}, it should look like:



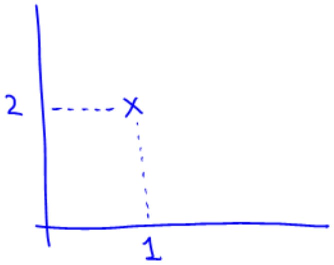
PROBLEM 2. (14 pts) Sparse Images and Vector Geometry

So far we've usually represented images as data grids. However, sometimes this can be very expensive in cases where most of the image pixels are exactly identical. For example, consider a 5X5 image where all the pixels are white, except for the pixel at position [2 3] which is black. Such an image is called a "sparse" image, and it may be much more efficient to represent it by only specifying the location(s) of the white pixel(s).

For example, if we had a 2 x 2 image with a white background, an image comprised of the vector $\begin{bmatrix} x & y \end{bmatrix}^T = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ would have a black pixel in the upper left hand corner of the image. (In other words, in this encoding the vectors correspond to positions in $\mathbb{R}^2$, so $\begin{bmatrix} 0 & 0 \end{bmatrix}^T$ is the origin.) This particular image is shown below. Note that for the rest of the problem, when we ask you to draw an image, rather than you filling in each pixel, we will just be asking you to draw an X at the positions indicated by each vector – this is depicted on the right for the example image.

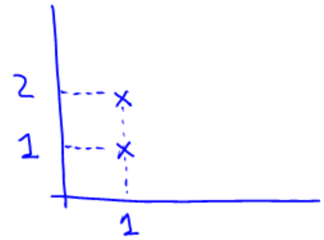


a) **(2 pts)** Using the same representation as the example described above, draw the image represented by the vector $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

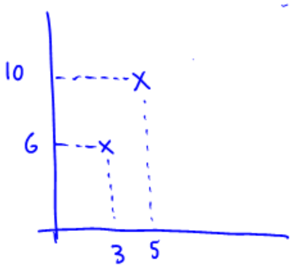


b) **(6 pts)** Now let's examine what happens if we have an image with two black pixels and perform a transformation on this image. In particular, let's assume that our image consists of the vectors $\begin{bmatrix}1\\2\end{bmatrix}$ and $\begin{bmatrix}1\\1\end{bmatrix}$. Draw both the original image and the image created by transforming that image by the matrix $\begin{bmatrix}1&2\\2&4\end{bmatrix}$. (Recall that transformation in this context means multiplying each vector by the given matrix.)

Original:



Transformed image:



$$\begin{bmatrix}1&2\\2&4\end{bmatrix}\cdot\begin{bmatrix}1\\2\end{bmatrix}=\begin{bmatrix}5\\10\end{bmatrix}$$

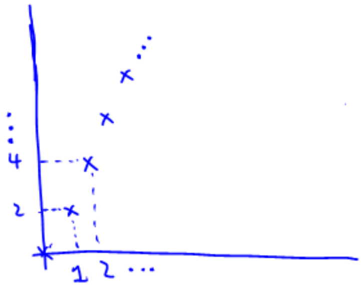
$$\begin{bmatrix}1&2\\2&4\end{bmatrix}\cdot\begin{bmatrix}1\\1\end{bmatrix}=\begin{bmatrix}3\\6\end{bmatrix}$$

c) **(6 pts)** Now let's imagine we had an image with black pixels everywhere, and that we transformed this image by the same $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$ matrix as in part b). Sketch the resulting transformed image. Note that you don't need to know the exact size of the original image – just indicate what happens to the transformed image as this size is increased. It may help you to consider what happens when you multiply an arbitrary vector $[x \;\; y]^T$ by the transformation matrix. What is property of the transformation matrix makes the resulting image look the way it does?

Notice that after transformation both of the black pixels in part b ended up on a single line. We can see this in general because:

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+2y \\ 2x+4y \end{bmatrix} = \begin{bmatrix} x+2y \\ 2\cdot(x+2y) \end{bmatrix} = \begin{bmatrix} a_1 \\ 2a_1 \end{bmatrix}$$

Therefore, an image with all black pixels will become a line defined by $y = 2x$:



The property of the matrix that causes this is that it is non-invertible.

PROBLEM 3. Coding for Storage (13 points)

Alice is a budding young photographer. Given the beautiful weather over the weekend she goes up into the Berkeley hills and takes a bunch of gorgeous pictures of San Francisco. She's hoping to sell these pictures to make some extra money, and so she wants to make sure she doesn't lose the data. The images she captures are all square images of *n* x *n* pixels. Each pixel contains a positive number indicating the intensity of light, and for the rest of this problem, you can assume that it takes 3 bytes of memory to store such a number.

a) **(2 pts)** Alice is known to be clumsy with her laptop and so she doesn't want to just store all the pictures in one place. Instead, she makes 3 copies of the image, and stores each copy on a different USB stick. If Alice wanted to store 1000 images this way, where each image is 1000 x 1000 pixels (i.e., *n* = 1000), how many total bytes of memory (including all three USB sticks) would Alice need?

$$N_{bytes} = 3 \text{ copies} \cdot 1000 \text{ images} \cdot (1000 \cdot 1000) \text{ pixels} \cdot 3 \text{ bytes/pixel}$$

$$= \boxed{9 \cdot 10^9 \text{ bytes}}$$

b) **(5 pts)** Alice realizes that if she wants to keep taking pictures, she is going to have to become efficient with her storage, she cannot simply keep replicating the data. So she focuses on first understanding how she can efficiently store two images. Just like in the lab, we'll express the images as vectors.

Consider two image vectors **x** and **y**. Alice applies the following strategy: she stores **x** on the laptop, **y** on the USB stick, and **x** on her phone. Alice's friend Elaine instead suggests that Alice use a different strategy and store **x**+**y** on her phone instead (while leaving the data stored on the laptop and USB the same as in Alice's strategy). Is Elaine's strategy better than Alice's strategy? If so, describe at least one scenario where Elaine's strategy can recover the data but Alice's cannot. If not, explain why the strategies are equivalent.

Elaine's strategy is better; in Alice's strategy, if the USB stick is lost, Alice can not recover y. With Elaine's strategy, any one of the storage devices can be lost while still recovering both x and y.

c) **(6 pts)** Let's now consider a more advanced storage strategy that uses 8 storage devices to handle four images **a**, **b**, **c**, and **d.** If each pixel (which contains a real number representing the intensity) in each of the four image vectors is a_i, b_i, c_i, d_i, the information stored in each pixel location of the 8 storage devices will be a_i, b_i, c_i, d_i, a_i+b_i, c_i+d_i, a_i+c_i, b_i+d_i. If we were to represent the storage action happening on each of the pixels as an 8x4 matrix multiplying a 4x1 vector, what would the entries in that 8x4 matrix and 4x1 vector be?

If the stored information is a vector s_i:

$$\begin{bmatrix} s_{i1} \\ s_{i2} \\ \vdots \\ \\ s_{i8} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix}$$

Define this matrix as Astore for part d).

d) **(Bonus: 6 pts)** Recalling that **a**, **b**, **c**, and **d** are all n^2 x 1 vectors, show how you could perform a single matrix-matrix multiplication to represent the storage action of all of the pixels simultaneously. Hint: what would be the dimensions of a matrix composed of [**a b c d**]$^{\mathbf{T}}$?

$$[a\;\; b\;\; c\;\; d]^T = \underset{\underset{N^2}{\longleftrightarrow}}{\begin{bmatrix} a^T \\ b^T \\ c^T \\ d^T \end{bmatrix}} \updownarrow 4$$

Since Astore is an 8x4 matrix, we can actually multiply the matrix above by Astore and get exactly what we wanted!

$$\begin{bmatrix} s_1^T \\ s_2^T \\ \vdots \\ s_8^T \end{bmatrix} = \text{Astore} \cdot \begin{bmatrix} a^T \\ b^T \\ c^T \\ d^T \end{bmatrix}$$

PROBLEM 4. (10 pts) Investment Strategy

Alice finally sells some of her photographs and makes a bunch of money. In fact, she is so rich that at the beginning of every day she can put \$1 of her money into the stock market. At the end of the day she sells her stocks and looks at how much money she got back. She decides to let a Berkeley-based mutual fund manage the \$1 she invests. This mutual fund invests in three companies from the area: Friendly Faces Inc., Search and Spend Inc., and Delicious Devices Inc.

Each company has a growth g_f(i), g_s(i), and g_d(i) on day *i*. What this means is that if Alice invests \$1 in Friendly Faces on day *i*, then she gets back g_f(i) dollars at the end of day *i*. The mutual fund invests fractions α_f, α_s, and α_d of Alice's money in each of the three companies; note that these fractions are always the same (i.e., they do not change from one day to the next), and that **all of her money is always invested in the market**.

a) **(3 pts)** Using the terms defined above, write an expression using a dot product that would compute how many dollars Alice gets back after the first day.

growth vector $\vec{g}(1) = \begin{bmatrix} g_f(1) \\ g_s(1) \\ g_d(1) \end{bmatrix}$

investment vector $\vec{\alpha} = \begin{bmatrix} \alpha_f \\ \alpha_s \\ \alpha_d \end{bmatrix}$

Dollars returned after day 1: $\vec{g}(1)^T \cdot \vec{\alpha} = \begin{bmatrix} g_f(1) & g_s(1) & g_d(1) \end{bmatrix} \begin{bmatrix} \alpha_f \\ \alpha_s \\ \alpha_d \end{bmatrix}$

$(= g_f(1)\alpha_f + g_s(1)\alpha_s + g_d(1)\alpha_d)$

b) **(7 pts)** Let's assume that on the first day, Alice got back \$1.7 from her \$1 investment – i.e., the result of the expression you wrote in part a) is \$1.7. Alice is quite happy with the increase, and decides to continue investing.

On the second day, Alice again invests \$1 at the beginning of the day, but at the end of the day she finds out that all of her money was lost – i.e., she will be getting \$0 back!

This makes Alice angry and she wants to find out the investment strategy of the mutual fund. She goes back and does some research and finds that $g_f(1) = 2$, $g_s(1) = 1$, and $g_d(1) = 2$, and on the second day $g_f(2) = 2$, $g_s(2) = 2$, but, $g_d(2) = -3$.

Write a matrix equation Alice could use to figure out what the mutual fund's investment strategy was (i.e., to find the values of α_f, α_s, and α_d).

* We know that $\vec{g}(1)^T \cdot \vec{\alpha} = 1.7$, and that $\vec{g}(2)^T \cdot \vec{\alpha} = 0$. This however gives us only two equations with 3 unknowns. To get a third equation, we need to remember that $\alpha_f + \alpha_s + \alpha_d = 1$ (i.e., that all of Alice's money is invested).

* Writing this in matrix form:

$$\begin{bmatrix} 2 & 1 & 2 \\ 2 & 2 & -3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_f \\ \alpha_s \\ \alpha_d \end{bmatrix} = \begin{bmatrix} 1.7 \\ 0 \\ 1 \end{bmatrix}$$

* So, assuming the matrix above is invertible (it is):

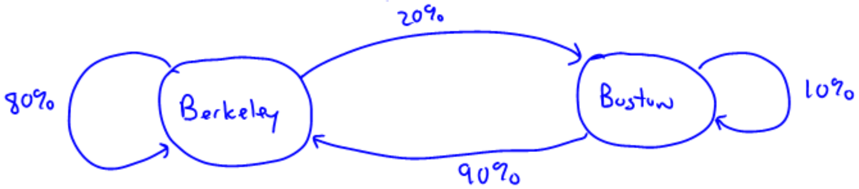
$$\begin{bmatrix} \alpha_f \\ \alpha_s \\ \alpha_d \end{bmatrix} = \begin{bmatrix} 2 & 1 & 2 \\ 2 & 2 & -3 \\ 1 & 1 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1.7 \\ 0 \\ 1 \end{bmatrix}$$

PROBLEM 5. (16 pts) Berkeley Forever!

Berkeley, as we all know, is a great place to live. But sometimes people make silly decisions, and move to places on the east coast (say Boston) and get stuck in blizzards. But of course once they move to the east coast, they quickly realize the futility of the endeavor and try to come back to Berkeley as soon as possible. We would like to understand the steady state populations in Beautiful Berkeley versus Blizzardy Boston. Let us assume that people only move back and forth between Berkeley and Boston and don't move anywhere else.

Every month, people make the decision to either stay in the same place or move to a new place. In a given month, 80% will continue to stay in Berkeley, but there is a 20% chance that they will choose to move to Boston. For every person in Boston, there is only a 10% chance they will choose to stay in Boston, but 90% will choose to move to Berkeley.

a) **(3 pts)** Draw a graph that models this setup. The nodes in the graph are Berkeley and Boston, and the arcs between the two nodes should be annotated with the probabilities of moving from one node to the other.



b) **(4 pts)** Now let's say we set up a vector $x(i) = \left[x_{berkeley}(i) \;\; x_{boston}(i) \right]^{\mathbf{T}}$, where $x_{\text{berkeley}}(i)$ represents the number of people living in Berkeley at month i (similarly for Boston and $x_{boston}(i)$). Write a matrix $A_{\text{transition}}$ that you could multiply by this vector to predict the number of people living in each city at month i+1 (i.e., $x(i+1) = A_{transition} x(i)$)

$$A_{transition} = \begin{bmatrix} 0.8 & 0.9 \\ 0.2 & 0.1 \end{bmatrix}$$

c) **(9 pts)** Now let us say the total population stays constant, and we start with half the population in Berkeley and the other half in Boston. Use the matrix you wrote above to find the steady-state populations in Berkeley and Boston. If the total population is 100, calculate the number of people in Berkeley at steady state?

* Best way to do this is to find the eigenvector of $A_{transition}$ that has an eigenvalue of 1 (we'll next find that there is one).
* Let's first find the eigenvalues by setting $\det(A_{transition} - \lambda I) = 0$

$$\det(A_{transition} - \lambda I) = (0.8-\lambda)(0.1-\lambda) - 0.2 \cdot 0.9$$

$$= 0.08 - 0.9\lambda + \lambda^2 - 0.18$$

$$= \lambda^2 - 0.9\lambda - 0.1 = 0$$

$$\hookrightarrow \lambda_{1,2} = \frac{0.9 \pm \sqrt{0.81 + 0.4}}{2}$$

$$= \frac{0.9 \pm \sqrt{1.21}}{2} = \frac{0.9 \pm 1.1}{2}$$

$$\hookrightarrow \lambda_1 = 1 \qquad \lambda_2 = -0.1$$

* Now let's find the eigenvector:

$$A_{transition} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\begin{aligned} 0.8v_1 + 0.9v_2 &= v_1 \\ 0.2v_1 + 0.1v_2 &= v_2 \end{aligned} \longrightarrow \begin{aligned} 0.2v_1 &= 0.9v_2 \\ v_1 &= 4.5v_2 \end{aligned}$$

Remember that total population is constant, so we should renormalize the vector such that $v_1 + v_2 = 1$

$$\hookrightarrow \text{Hence: } \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{4.5}{5.5} \\ \frac{1}{5.5} \end{bmatrix}$$

So if we started with 100 people, we will end up with ~ 82 people in Berkeley, and ~ 18 people in Boston.