EECS 16A Designing Information Devices and Systems I
Spring 2015 Homework 1

This homework is due February 5, 2015 at 5PM.

1. Imaging lab

Download the file `Homework1.ipynb` and use the code to do the following.

(a) Generate a black pixel at $(0,0)$ in a 6 pixel $\times$ 6 pixel grid. $(0,0)$ is the upper left corner of the grid.

(b) Generate a black pixel at location $(2,3)$ in a 6 pixel $\times$ 6 pixel grid.

(c) Generate a figure that is black on $(0:2,:)$ and white on $(3:5,:)$.

(d) Output the data-grid as a vector. To unwrap a matrix, stack the rows next to each other and take the transpose to get a column vector. For example if

$$A_{\text{matrix}} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$$

then, $A_{\text{vector}} = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{2,1} & A_{2,2} \end{bmatrix}^T$

2. Inner products

The Cauchy-Schwarz inequality states that for two vectors $x, y \in \Re^n$:

$$|x^T y| \leq ||x|| \cdot ||y||$$

Use the Cauchy-Schwarz inequality to verify the triangle inequality:

$$||x+y|| \leq ||x|| + ||y||$$

3. Audio file matching

Lots of different quantities we interact with everyday can be expressed as vectors. For example, an audio clip can be thought of as a vector. The series of numbers in the clip determine the sounds we hear. An audio segment or a sound wave is a continuous function of time, but as we saw in the last homework this can be sampled at regular intervals to make a discrete sequence of numbers that can be represented as a vector. The real value recorded after the signal is converted to discrete samples that can then be quantized to give a sequence of "bits" which can be used to approximate the real number. We will not be going into detail about this process here, but instead will consider simplified version of an audio signal.

Let us consider a very simplified model for an audio signal, one that is just composed of two tones. One is represented by -1 and the other by $+1$. A vector of length n makes up the audio file.

(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 $X_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$ and $X_2 = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$? What is the dot product of these two vectors? What if $X_1 = \begin{bmatrix} 1 & 1 \cdots 1 \end{bmatrix}^T$ and $X_2 = \begin{bmatrix} -1 & -1 & \cdots & -1 \end{bmatrix}^T$? Can you come up with an idea to compare to general vectors of length n now?

(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*, to be able to identify a song from a signature tune. Consider the vector of length 8, $X = \begin{bmatrix}-1 & 1 & 1 & -1 & 1 & 1 & -1 & -1\end{bmatrix}^T$. Let us label the elements of X so that $X = \begin{bmatrix}x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8\end{bmatrix}^T$. We want to find the short segment $Y = \begin{bmatrix}1 & 1 & 1\end{bmatrix}^T$ in the longer vector, i.e., we want to find i, such that the sequence represented by $\begin{bmatrix}x_i & x_{i+1} & x_{i+2}\end{bmatrix}$ is the closest to Y. How can we find this? Applying the same technique what i gives the best match for $Y = \begin{bmatrix}1 & -1 & 1\end{bmatrix}^T$?

(c) (Bonus) Now suppose our vector was represented using integers and not just by 1 and -1. Say we wanted to locate the sequence closest to $Y = \begin{bmatrix}1 & 2 & 3\end{bmatrix}^T$ in $X = \begin{bmatrix}1 & 2 & 3 & 4 & 5 & 6 & 7 & 8\end{bmatrix}^T$. What happens if you apply the technique of part (b)? How would you modify this technique for the problem here?

4. Image Blending

Blending gray-scale images can be modeled as a linear combination of image vectors. In this question, we will blend sections of two images in different ratios using vector scaling and addition. Download `cal.npy` and `campanile.npy` in the same folder as `Homework1.ipynb` before running the code.

(a) Notice that both these variables are the vector form of image data-grids. Blend the two images by scaling each by vector by 0.5 and adding the results. **Make sure to model the vector scaling as a matrix multiplication!** Finally, display the hybrid image.

(b) Now, blend the images in the following manner:

Top half of Final Image = (0.8*Top half of Cal) + (0.2*Top half of Campanile)
Bottom half of Final Image = (0.4*Bottom half of Cal) + (0.6*Bottom half of Campanile)

You should only use one scaling matrix per image vector! Again, display the hybrid image.

(c) Say the only vector provided was a blend of the Cal and Campanile images. Assume $Cl_{i,j}$ and $Cm_{i,j}$ represent the pixel value in the i^{th} row and j^{th} column of the Cal and Campanile images, respectively. The given vector has the form

$$\begin{bmatrix} Cm_{1,1} \\ Cl_{1,1} \\ Cm_{1,2} \\ Cl_{1,2} \\ Cm_{1,3} \\ \vdots \end{bmatrix}$$

With this particular vector, it is possible to perform the same operation as part (b) with **a single matrix multiplication operation**. What would this matrix look like? Why? (You don't need to code the matrix for this part. Explain the form of the matrix and write out its first few rows and columns).