This homework is due October 5 2015, at Noon.

1. **Mechanical Problems**

   (a) Compute the determinant of $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$

   Draw what this matrix does to the unit square in 2D. Compute the area of the resulting shape using standard geometric arguments.

   (b) Compute the determinant of $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$

   Draw what this matrix does to the unit square in 2D. Compute the area of the resulting shape using standard geometric arguments.

   (c) Compute the determinant of $\begin{bmatrix} -4 & 0 & 0 & 0 \\ 0 & 17 & 0 & 0 \\ 0 & 0 & -31 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$

   (d) Find the eigenvalues and the eigenspaces of the following matrix A:

   $A = \begin{bmatrix} 2 & 3 \\ 3 & 4 \end{bmatrix}$

   What is the inner product between the two eigenvectors?

2. **Image Compression**

   In this question, we explore how eigenvalues and eigenvectors can be used for image compression. We have seen that a grayscale image can be represented as a data grid. Say a symmetric, square image is represented by a symmetric matrix A, such that $A^T = A$. We've been transforming the images to vectors in the past to make it easier to process them as data, but here we will understand them as 2D data. Let $\lambda_1 \cdots \lambda_n$ be the eigenvalues of A with corresponding eigenvectors $v_1 \cdots v_n$. Then, the matrix can be represented as

   $$A = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_n v_n v_n^T$$

   However, the matrix A can also be *approximated* with the k largest eigenvalues and corresponding eigenvectors. That is,

   $$A \approx \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T + \cdots + \lambda_k v_k v_k^T$$

   (a) Can you construct appropriate matrices U, V (using v_i's as rows and columns) and a matrix Λ with the eigenvalues λ_i as components such that

   $$A = U \Lambda V$$

   .

(b) Use the IPython notebook `prob5.ipynb` and the image file `pattern.npy`. Use the `numpy.linalg` command `eig` to find the U and Λ matrices for the image. Mathematically, how many eigenvectors are required to fully capture the information within the image?

(c) In the IPython notebook, find an approximation for the image using the 100 largest eigenvalues and eigenvectors.

(d) Repeat part (c) with $k = 50$. By further experimenting with the code, what seems to be the lowest value of k that retains most of the salient features of the given image?

3. Counting the paths of a Random Surfer

In class, we discussed the behavior of a random web-surfer who jumps from webpage to webpage. We would like to know how many possible paths there are for a random surfer to get from a page to another page. To do this, we represent the webpages as a graph. If page 1 has a link to page 2, we have a directed edge from page 1 to page 2. This graph can further be represented by what is known as an "adjacency matrix", A, with elements a_{ij}. $a_{ji} = 1$ if there is link from page i to page j. Matrix operations on the adjacency matrix make it very easy to compute the number of paths to get from one webpage to webpage.

This path counting actually is an implicit part of the how the "importance scores" for each webpage are described. Recall that the "importance score" of a website is the steady-state frequency of the fraction of people on that website.

Consider the following graphs.

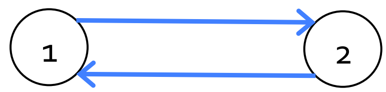

Figure 1: Graph A

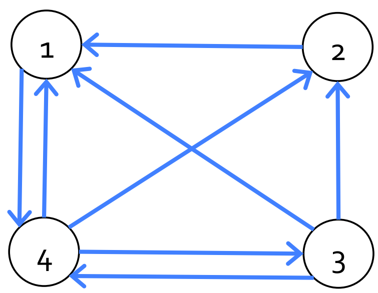

Figure 2: Graph B

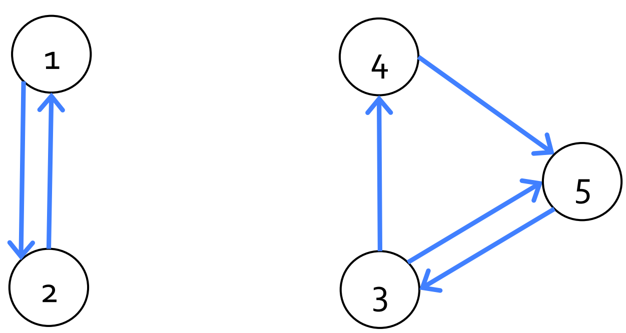

Figure 3: Graph C

(a) Write out the adjacency matrix for graph A.

(b) For graph A: How many one-hop paths are there from webpage-1 to webpage-2? How many two-hop paths are there from webpage-1 to webpage-2? How about 3-hop?

(c) For graph A: What are the importance scores of the two webpages?

(d) Write out the adjacency matrix for graph B.

(e) For graph B: How many two-hop paths are there from webpage-1 to webpage-3? How many three-hop paths are there from webpage-1 to webpage-2?

(f) For graph B: What are the importance scores of the webpages?

(g) Write out the adjacency matrix for graph C.

(h) For graph C: How many paths are there from webpage-1 to webpage-3?

(i) For graph C: What are the importance scores of the webpages? How is graph (c) different from graph (b), and how does this relate the importance scores and eigenvalues and eigenvectors you found?

4. Row Operations and Determinants

In this question we explore the effect of row operations on the determinant of a matrix. Recall from lecture that scaling a row by a will increase the determinant by a, and adding a multiple of one row to another will not change the determinant. The determinant of an identity matrix is 1 (corresponding to the volume of a unit hypercube).

(a) An upper triangular matrix is a matrix with zero below its diagonal. For example a 3×3 upper triangular matrix is :

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ 0 & b_2 & b_3 \\ 0 & 0 & c_3 \end{bmatrix}$$

By considering row-operations and what they do to a determinant, argue that the determinant of a general $n \times n$ upper-triangular matrix is the product of its diagonal entries, if they are non-zero. For example, the determinant of the 3×3 matrix above is $a_1 \times b_2 \times c_3$ if $a_1, b_2, c_3 \neq 0$.

(b) If the diagonal of an upper-triangular matrix has a zero entry, argue that its determinant is still the product of its diagonal entries.

(c) Find a formula for the determinant of a general 3×3 matrix using Gaussian elimination, by keeping track of what each row operation does to the determinant of the matrix while reducing it to the identity matrix.

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}$$

You may assume that the matrix is structured so that no division by zero occurs in your calculations (this simplifies the proof at the expense of getting a less general result). After simplification, you should get a summation with 6 terms.

(d) Do you see a pattern in the formula of the 3×3 determinant you derived in the previous part? How many times do the row indices a, b, c appear in each term of the determinant? How many times do the column indices 1, 2, 3 appear in each term of the determinant?

5. Sports Rank

Every year in College sports, specifically football and basketball, debate rages over team rankings. The rankings determine who will get to compete for the ultimate prize, the national championship. However, ranking teams is quite challenging in the setting of college sports for a few reasons: there is uneven paired competition (not every team plays each other), sparsity of matches (every team plays a small subset of all the teams available), and there is no well-ordering (team A beats team B who beats team C who beats A). In this problem we will come up with an algorithm to rank the teams, with real data drawn from the 2014 Associated Press (AP) top 25 College football teams.

Given N teams we want to determine the rating r_i for the i^{th} team for $i = 1, 2, \ldots, N$, after which the teams can be ranked in order from highest to lowest rating. Given the wins and losses of each team we can assign each team a score s_i.

$$s_i = \sum_{j}^{N} q_{ij} r_j, \tag{1}$$

where q_{ij} represents the number of times team i has beaten team j divided by the number [1] of games played by team i. If we define the vectors $\vec{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix}$, and $\vec{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}$ we can express there relationship as a system of equations

$$\vec{s} = Q\vec{r}, \tag{2}$$

where $Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1} & q_{N2} & \cdots & q_{NN} \end{bmatrix}$ is an $N \times N$ matrix.

(a) Consider a specific case where we have three teams, team A, team B, and team C. Team A beats team C twice and team B once. Team B beats team A twice and never beats team C. Team C beats team B three times. What is the matrix Q?

(b) Returning to the general setting, if our scoring metric is good, then it should be the case that teams with better ratings have higher scores. Let's make the assumption that $s_i = \lambda r_i$ with $\lambda > 0$. Show that $\vec{r}$ is an eigenvector of Q.

To find our rating vector we need to find an eigenvector of Q with all nonnegative entries (ratings can't be negative) and a positive eigenvalue. If the matrix Q satisfies certain conditions (beyond the scope of this course) the dominant eigenvalue λ_D, i.e. the largest eigenvalue in absolute value, is positive and real. In addition, the dominant eigenvector, i.e. the eigenvector associated with the dominant eigenvalue, is unique and has all positive entries. We will now develop a method for finding the dominant eigenvector for a matrix when it is unique.

(c) Given $\vec{v}$ is an eigenvector of Q with eigenvalue λ and a real nonzero number c, express $Q^n c\vec{v}$ in terms of $\vec{v}$, c and λ

[1] We normalize by the number of games played to prevent teams from getting a high score by just repeatedly playing against weak opponents

(d) Now given multiple eigenvectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m$ of Q, their eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$, and real nonzero numbers $c_1, c_2, \ldots, c_m$, express $Q^n(\sum_{i=1}^m c_i \vec{v}_i)$ in terms of $\vec{v}$'s, λ's, and c's.

(e) Assuming that $|\lambda_1| > |\lambda_i|$ for $i = 2, \ldots, m$, argue or prove

$$\lim_{n\to\infty} \frac{1}{\lambda_1^n} Q^n\left(\sum_{i=1}^m c_i \vec{v}_i\right) = c_1 \vec{v}_1 \tag{3}$$

Hints:

i. For sequences of vectors $\{\vec{a}_n\}$ and $\{\vec{b}_n\}$, $\lim_{n\to\infty}(\vec{a}_n + \vec{b}_n) = \lim_{n\to\infty} \vec{a}_n + \lim_{n\to\infty} \vec{b}_n$.

ii. For a scalar w with $|w| < 1$, $\lim_{n\to\infty} w^n = 0$.

(f) Now further assuming that λ_1 is positive show

$$\lim_{n\to\infty} \frac{Q^n(\sum_{i=1}^m c_i \vec{v}_i)}{||Q^n(\sum_{i=1}^m c_i \vec{v}_i)||} = \frac{c_1 \vec{v}_1}{||c_1 \vec{v}_1||} \tag{4}$$

Hints:

i. Divide the numerator and denominator by λ_1^n and use the result from the previous part.

ii. For the sequence of vectors $\{\vec{a}_n\}$, $\lim_{n\to\infty} ||\vec{a}_n|| = ||\lim_{n\to\infty} \vec{a}_n||$.

iii. For a sequence of vectors $\{\vec{a}_n\}$ and a sequence of scalars $\{\alpha_n\}$, if $\lim_{n\to\infty} \alpha_n$ is not equal to zero then the $\lim_{n\to\infty} \frac{\vec{a}_n}{\alpha_n} = \frac{\lim_{n\to\infty} \vec{a}_n}{\lim_{n\to\infty} \alpha_n}$.

Let's assume that any vector $\vec{b}$ in $\mathbb{R}^N$ can be expressed as a linear combination of the eigenvectors of any square matrix A in $\mathbb{R}^{N\times N}$, i.e. A has N rows and N columns.

Let's tie it all together. Given the eigenvectors of Q, $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N$, we arbitrarily choose the dominant eigenvector to be $\vec{v}_1 = \vec{v}_D$. If we can find a vector $\vec{b} = \sum_{i=1}^m c_i \vec{v}_i$, such that c_1 is nonzero, then [2]

$$\lim_{n\to\infty} \frac{Q^n \vec{b}}{||Q^n \vec{b}||} = \frac{c_1 \vec{v}_D}{||c_1 \vec{v}_D||}. \tag{5}$$

This is the idea behind the power iteration method, which is a method for finding the unique dominant eigenvector (up to a scalar) of a matrix whenever one exists. In the Ipython notebook we will use this method to rank our teams. Note: For this application we know the rating vector (which will be the dominant eigenvector) has all positive entries, but c_1 might be negative resulting in our method returning a vector with all negative entries. If this happens, we simply multiply our result by -1.

(g) From the method you implemented in the Ipython notebook name the top five teams, the fourteenth team, and the seventeenth team.

6. Your Own Problem Write your own problem related to this week's material and solve it. You may still work in groups to brainstorm problems, but each student should submit a unique problem. What is the problem? How to formulate it? How to solve it? What is the solution?

[2] If we select a vector at random c_1 will be nonzero almost certainly.