EECS 16A Fall 2016 — Designing Information Devices and Systems I
Babak Ayazifar, Vladimir Stojanovic

Homework 13

This homework is due December 6, 2016, at 13:00.

1. **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.

   Solution: I worked on this homework with...

   I first worked by myself for 2 hours, but got stuck on Problem 5 so I went to office hours on...

   Then I went to homework party for a few hours, where I finished the homework.

2. **Mechanical Problem**

   Compute the eigenvalues and eigenvectors of the following matrices.

   (a) $\begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix}$

   Solution: There are two ways to do this. First, by inspection. We can see that this matrix multiplies everything in the first coordinate by 3 and everything in the second by 5. Consequently, when fed $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ it will return 5 times the input. And when fed $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ it will return 3 times the input vector.

   Alternatively, we can use determinants.

$$\begin{vmatrix} 3-\lambda & 0 \\ 0 & 5-\lambda \end{vmatrix} = 0$$

$$(3-\lambda)(5-\lambda) - 0 = 0$$

   This is already factored for you! We see that, by definition, diagonal matrices have their eigenvalues on the diagonal.

   $\lambda = 3:$

$$AX = 3X \implies (A - 3I_3)X = 0$$

$$\begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix} - \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} = 0 \implies \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix} = 0 \implies y = 0 \implies \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

   $\lambda = 5:$

$$\begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix} = 0 \implies x = 0 \implies \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

(b) $\begin{bmatrix} 22 & 6 \\ 6 & 13 \end{bmatrix}$

Solution: Here it is hard to guess the answers.

$$(22-\lambda)(13-\lambda)-36=0 \implies \lambda = 10, 25$$

$\lambda = 10:$

$$\begin{bmatrix} 12 & 6 \\ 6 & 3 \end{bmatrix} \implies \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} \implies 2x+y=0 \implies \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$\lambda = 25:$

$$\begin{bmatrix} -3 & 6 \\ 6 & -12 \end{bmatrix} \implies \begin{bmatrix} -1 & 2 \\ 0 & 0 \end{bmatrix} \implies 2y = x \implies \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

(c) $\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$

Solution: This can also be seen by inspection. The matrix is clearly not invertible since the first two rows are linearly dependent. So, there must be a 0 eigenvalue. This has eigenvector $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$.

The other eigenvector can be seen by noticing that the second row is twice the first. So $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ is a good guess to try and indeed it works with $\lambda = 5$.
Alternatively, we can brute force it.

$$\begin{vmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{vmatrix} = 0$$

$$\begin{aligned} (1-\lambda)(4-\lambda)-4 &= 0 \\ \lambda^2 - 5\lambda = 0 &\implies \lambda(\lambda-5) = 0 \\ \lambda &= 0, 5 \end{aligned}$$

$\lambda = 0:$

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} = 0 \implies \begin{bmatrix} 1 & 2 \\ 0 & 0 \end{bmatrix} = 0$$

$$x + 2y = 0 \implies y \; -\frac{1}{2}x \implies \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

$\lambda = 5:$

$$\begin{bmatrix} 1-5 & 2 \\ 2 & 4-5 \end{bmatrix} = \begin{bmatrix} -4 & 2 \\ 2 & -1 \end{bmatrix} = 0 \implies \begin{bmatrix} 2 & -1 \\ 0 & 0 \end{bmatrix} = 0$$

$$2x - y = 0 \implies y = 2x \implies \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

(d) $\begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$ (What special matrix is this?)

Solution: This is clearly a rotation matrix (clockwise by 30 degrees) because we notice the pattern. Diagonals matching and off-diagonals being inverses of each other. So it does not have any real eigenvalues or eigenvectors.

We can still brute-force it however to get complex ones.

$$\left(\frac{\sqrt{3}}{2} - \lambda\right)^2 + \frac{1}{4} = 0 \implies \lambda = \frac{\sqrt{3} \pm i}{2}$$

$\lambda = \frac{\sqrt{3}+i}{2}$:

$$\begin{bmatrix} \frac{-i}{2} & \frac{-1}{2} \\ \frac{1}{2} & \frac{-i}{2} \end{bmatrix} \implies \begin{bmatrix} -i & -1 \\ 1 & -i \end{bmatrix} \implies \begin{bmatrix} 1 & -i \\ 0 & 0 \end{bmatrix}$$

$$x - iy = 0 \implies \begin{bmatrix} i \\ 1 \end{bmatrix}$$

$\lambda = \frac{\sqrt{3}-i}{2}$:

$$\begin{bmatrix} \frac{i}{2} & \frac{-1}{2} \\ \frac{1}{2} & \frac{i}{2} \end{bmatrix} \implies \begin{bmatrix} i & -1 \\ 1 & i \end{bmatrix} \implies \begin{bmatrix} 1 & i \\ 0 & 0 \end{bmatrix}$$

$$x + iy = 0 \implies \begin{bmatrix} 1 \\ i \end{bmatrix}$$

3. Mechanical Diagonalization

All calculations in this problem are intended to be done by hand, but you can use a computer to check your work.

Diagonalize the matrix

$$A = \begin{bmatrix} 1/2 & 1/2 & -1/2 \\ -1/2 & 3/2 & 1/2 \\ -1 & 1 & 1 \end{bmatrix} \tag{1}$$

given that A has eigenvalues 1, 2, and 0.

Solution:

First we compute the eigenvectors of A given the eigenvalues.

For the eigenvalue of 2 the eigenvector spans

$$\text{Nullspace of}\left(\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} - \begin{bmatrix} 1/2 & 1/2 & -1/2 \\ -1/2 & 3/2 & 1/2 \\ -1 & 1 & 1 \end{bmatrix}\right) = \text{Nullspace of}\left(\begin{bmatrix} 3/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \\ 1 & -1 & 1 \end{bmatrix}\right) \tag{2}$$

By inspection we get the eigenvector is $\begin{bmatrix} 0, & 1, & 1 \end{bmatrix}^T$. (If we didn't notice that the summing the last two columns gives 0, we could have solved using Gaussian Elimination.)

For the eigenvalue of 1 the eigenvector spans

$$\text{Nullspace of}\left(\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}-\begin{bmatrix}1/2&1/2&-1/2\\-1/2&3/2&1/2\\-1&1&1\end{bmatrix}\right)=\text{Nullspace of}\left(\begin{bmatrix}1/2&-1/2&1/2\\1/2&-1/2&-1/2\\1&-1&0\end{bmatrix}\right) \tag{3}$$

By inspection, again we get that the eigenvector is $[\ 1,\ 1,\ 0\ ]^T$.

For the eigenvalue of 0, the eigenvector simply spans the nullspace of A.

$$\text{Nullspace of}\left(\begin{bmatrix}1/2&1/2&-1/2\\-1/2&3/2&1/2\\-1&1&1\end{bmatrix}\right) \tag{4}$$

Again by inspection, we get the eigenvector is $[\ 1,\ 0,\ 1\ ]^T$.

We now write a matrix P whose columns are the eigenvectors of A.

$$P=\begin{bmatrix}0&1&1\\1&1&0\\1&0&1\end{bmatrix} \tag{5}$$

We compute P^{-1} using Gaussian Elimination

$$\left[\begin{array}{ccc|ccc}1&0&0&0&1&1\\0&1&0&1&1&0\\0&0&1&1&0&1\end{array}\right]\underbrace{\Rightarrow}_{\substack{\text{Switching}\\\text{row order}}}\left[\begin{array}{ccc|ccc}0&0&1&1&0&1\\0&1&0&1&1&0\\1&0&0&0&1&1\end{array}\right]\underbrace{\Rightarrow}_{R2\leftarrow R2-R1}\left[\begin{array}{ccc|ccc}0&0&1&1&0&1\\0&1&-1&0&1&-1\\1&0&0&0&1&1\end{array}\right] \tag{6}$$

$$\underbrace{\Rightarrow}_{\substack{R3\leftarrow R3-R2\\R3\leftarrow\frac{1}{2}R3}}\left[\begin{array}{ccc|ccc}0&0&1&1&0&1\\0&1&-1&0&1&-1\\1/2&-1/2&1/2&0&0&1\end{array}\right]\underbrace{\Rightarrow}_{\substack{R1\leftarrow R1-R3\\R2\leftarrow R2+R3}}\left[\begin{array}{ccc|ccc}-1/2&1/2&1/2&1&0&0\\1/2&1/2&-1/2&0&1&0\\1/2&-1/2&1/2&0&0&1\end{array}\right] \tag{7}$$

Thus

$$P^{-1}=\begin{bmatrix}-1/2&1/2&1/2\\1/2&1/2&-1/2\\1/2&-1/2&1/2\end{bmatrix} \tag{8}$$

(It's a good idea to check that $PP^{-1}=I$ which they do.)

We can now write A in it's diagonal form as

$$A=PDP^{-1}=\begin{bmatrix}0&1&1\\1&1&0\\1&0&1\end{bmatrix}\begin{bmatrix}2&0&0\\0&1&0\\0&0&0\end{bmatrix}\begin{bmatrix}-1/2&1/2&1/2\\1/2&1/2&-1/2\\1/2&-1/2&1/2\end{bmatrix} \tag{9}$$

(It's a good idea to multiply it out to check that you get A.)

4. Spectral Mapping and the Fibonacci Sequence

One of the most useful things about diagonalization is it allows us to easily compute polynomial functions of matrices. This in turn lets us do far more, including solving many linear recurrence relations. This problem shows you how this can be done for the Fibonacci numbers, but you should notice that the same exact technique can apply far more generally.

Suppose we have a matrix A that can be diagonalized as

$$A = PDP^{-1} = \begin{bmatrix} | & & | \\ \vec{p}_1 & \cdots & \vec{p}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} | & & | \\ \vec{p}_1 & \cdots & \vec{p}_n \\ | & & | \end{bmatrix}^{-1} \tag{10}$$

where D is a diagonal matrix with the eigenvalues $\lambda_1, \ldots, \lambda_n$ on the diagonal and P is a matrix whose columns $\vec{p}_1, \ldots, \vec{p}_n$ are the eigenvectors of A.

(a) **Write out A^N in terms of P,P^{-1}, and D and simplify it as much as you can.** You should be able to show that you can write A^N as

$$A^N = PD^NP^{-1} = \begin{bmatrix} | & & | \\ \vec{p}_1 & \cdots & \vec{p}_n \\ | & & | \end{bmatrix} \begin{bmatrix} \lambda_1^N & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \lambda_n^N \end{bmatrix} \begin{bmatrix} | & & | \\ \vec{p}_1 & \cdots & \vec{p}_n \\ | & & | \end{bmatrix}^{-1} \tag{11}$$

What does this say about any polynomial function of A?

Solution:

$$A^N = \underbrace{A \times A \times \cdots \times A}_{\times N} \tag{12}$$

$$= PD\underbrace{P^{-1} \times P}_{I}D\underbrace{P^{-1} \times P}_{I}DP^{-1} \times \cdots \times PD\underbrace{P^{-1} \times P}_{I}DP^{-1} \tag{13}$$

$$= PD^NP^{-1} \tag{14}$$

Since D is diagonal with the eigenvalues of $\lambda_1, \ldots, \lambda_n$ on the diagonal, we can easily compute

$$D^N = \begin{bmatrix} \lambda_1^N & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & \lambda_n^N \end{bmatrix} \tag{15}$$

We note that this implies that we can evaluate any polynomial function of A by simply applying that polynomial to the eigenvalues of A. For example, can write

$$3A^{10} - 2A^9 = P\left(3D^{10}\right)P^{-1} + P\left(-2D^9\right)P^{-1} \tag{16}$$

$$= P\left(3D^{10} - 2D^9\right)P^{-1} \tag{17}$$

$$= P \begin{bmatrix} 3\lambda_1^{10} - 2\lambda_1^9 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 3\lambda_n^{10} - 2\lambda_n^9 \end{bmatrix} P^{-1} \tag{18}$$

and similarly for any polynomial.

(b) This idea that for diagonalizable matrices you can raise a matrix to any power by simply raising it's eigenvalues to that power is part of the **spectral mapping theorem**. We will now illustrate the power of this theorem to compute analytical expressions for numbers in the famous Fibonacci sequence.

Take a look at the Wikipedia article and find a cool fact about Fibonacci numbers to report!

Solution:

Give yourself credit for any cool fact. One of our favorites is "This means that every positive integer can be written as a sum of Fibonacci numbers, where any one number is used once at most."

(c) The Fibonacci sequence can be constructed according to the following relation. The Nth number in the Fibonacci sequence, F_N is computed by adding the previous two numbers in the sequence together

$$F_N = F_{N-1} + F_{N-2} \tag{19}$$

We select the first two numbers in the sequence to be $F_1 = 0$ and $F_2 = 1$ and then we can compute the following numbers as

$$0,\ 1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ldots \tag{20}$$

Notice that we can write the operation of computing the next Fibonacci numbers from the previous two using matrix multiplication

$$\begin{bmatrix} F_N \\ F_{N-1} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}_{A} \begin{bmatrix} F_{N-1} \\ F_{N-2} \end{bmatrix} \tag{21}$$

Do you see why? Notice also that we could use powers of A to compute Fibonacci numbers starting from the original two, 0 and 1.

$$\begin{bmatrix} F_N \\ F_{N-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{N-2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{22}$$

Diagonalize A and use Equation (11) to show that

$$F_N = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{N-1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{N-1} \tag{23}$$

is an analytical expression for the Nth Fibonacci number.

Note that A has eigenvalues and eigenvectors

$$\left\{\lambda_1 = \frac{1+\sqrt{5}}{2},\ \lambda_2 = \frac{1-\sqrt{5}}{2}\right\} \qquad \left\{\vec{p}_1 = \begin{bmatrix} \frac{1+\sqrt{5}}{2} \\ 1 \end{bmatrix},\ \vec{p}_2 = \begin{bmatrix} \frac{1-\sqrt{5}}{2} \\ 1 \end{bmatrix}\right\} \tag{24}$$

Feel free to use the 2×2 inverse formula

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{25}$$

Solution: First, we diagonalize A

$$P = \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix} \qquad P^{-1} = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 & \frac{-1+\sqrt{5}}{2} \\ -1 & \frac{1+\sqrt{5}}{2} \end{bmatrix} \tag{26}$$

$$A = PDP^{-1} \tag{27}$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1+\sqrt{5}}{2} & 0 \\ 0 & \frac{1-\sqrt{5}}{2} \end{bmatrix} \left(\frac{1}{\sqrt{5}}\begin{bmatrix} 1 & \frac{-1+\sqrt{5}}{2} \\ -1 & \frac{1+\sqrt{5}}{2} \end{bmatrix}\right) \tag{28}$$

Then, we have that F_N is equal to the first element of $A^{N-2}\begin{bmatrix}1\\0\end{bmatrix}$.

$$F_N = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}^{N-2}\begin{bmatrix}1\\0\end{bmatrix} \tag{29}$$

$$= \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2}\\1 & 1\end{bmatrix}\begin{bmatrix}\frac{1+\sqrt{5}}{2} & 0\\0 & \frac{1-\sqrt{5}}{2}\end{bmatrix}^{N-2}\left(\frac{1}{\sqrt{5}}\begin{bmatrix}1 & \frac{-1+\sqrt{5}}{2}\\-1 & \frac{1+\sqrt{5}}{2}\end{bmatrix}\right)\begin{bmatrix}1\\0\end{bmatrix} \tag{30}$$

$$= \begin{bmatrix}\frac{1+\sqrt{5}}{2} & \frac{1-\sqrt{5}}{2}\end{bmatrix}\begin{bmatrix}\left(\frac{1+\sqrt{5}}{2}\right)^{N-2} & 0\\0 & \left(\frac{1-\sqrt{5}}{2}\right)^{N-2}\end{bmatrix}\begin{bmatrix}\frac{1}{\sqrt{5}}\\-\frac{1}{\sqrt{5}}\end{bmatrix} \tag{31}$$

$$= \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{N-1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{N-1} \tag{32}$$

(d) **(Bonus In-Scope)** Generalize what you found to a procedure that will give you, in principle, expressions for many linear recurrence relations that are recursively defined as $S_{n+k} = \sum_{i=0}^{k-1}\alpha_i S_{n+i}$ for some coefficients $\vec{\alpha}$ and initial conditions $[S_{k-1}, S_{k-2}, \ldots, S_0]^T = \vec{s}_0$.

Do this by setting up the appropriate matrix A and then invoking a computation of its eigenvalues and eigenvectors. And then using the results. (Feel free to assume diagonalizability of the resulting matrix, although there are some important cases when that does not hold.)

Solution:

Similar to Equation (21), we can write this recursive relationship as

$$\begin{bmatrix}S_{n+k}\\S_{n+k-1}\\S_{n+k-2}\\\vdots\\S_{n+1}\end{bmatrix} = \underbrace{\begin{bmatrix}\alpha_{k-1} & \alpha_{k-2} & \cdots & \alpha_1 & \alpha_0\\1 & 0 & \cdots & 0 & 0\\0 & 1 & \cdots & 0 & 0\\\vdots & & \ddots & & \vdots\\0 & 0 & \cdots & 1 & 0\end{bmatrix}}_{A}\begin{bmatrix}S_{n+k-1}\\S_{n+k-2}\\\vdots\\S_{n+1}\\S_n\end{bmatrix} \tag{33}$$

We can then write S_N (for $N > k-1$) as

$$S_N = \vec{e}_1^{\,T}A^{N-(k-1)}\vec{s}_0 \tag{34}$$

where $\vec{e}_1^{\,T} = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}$ We can write $A = PDP^{-1}$ where D is a diagonal matrix with the eigenvalues of A on the diagonal $(\lambda_1, \ldots, \lambda_n)$ and the columns of P are the corresponding eigenvectors of A, $(\vec{p}_1, \ldots, \vec{p}_n)$. The above expression then becomes

$$S_N = \vec{e}_1^{\,T}PD^{N-(k-1)}P^{-1}\vec{s}_0 \tag{35}$$

If we define

$$P = \begin{bmatrix}| & & |\\\vec{p}_1 & \cdots & \vec{p}_n\\| & & |\end{bmatrix} \qquad P^{-1} = \begin{bmatrix}- & \vec{q}_1^{\,T} & -\\ & \vdots & \\- & \vec{q}_n^{\,T} & -\end{bmatrix} \tag{36}$$

we can write out S_N explicitly as

$$S_N = \vec{e}_1^T \vec{p}_1 \lambda_1^{N-(k-1)} \vec{q}_1^T \vec{s}_0 + \cdots + \vec{e}_1^T \vec{p}_n \lambda_n^{N-(k-1)} \vec{q}_n^T \vec{s}_0 \tag{37}$$

$$= \sum_{i=1}^{n} \vec{e}_1^T \vec{p}_i \lambda_i^{N-(k-1)} \vec{q}_i^T \vec{s}_0 \tag{38}$$

5. 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 $\vec{v}_1 \cdots \vec{v}_n$. Then, the matrix can be represented as

$$A = \lambda_1 \vec{v}_1 \vec{v}_1{}^T + \lambda_2 \vec{v}_2 \vec{v}_2{}^T + \cdots + \lambda_n \vec{v}_n \vec{v}_n{}^T$$

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

$$A \approx \lambda_1 \vec{v}_1 \vec{v}_1{}^T + \lambda_2 \vec{v}_2 \vec{v}_2{}^T + \cdots + \lambda_k \vec{v}_k \vec{v}_k{}^T$$

(a) Can you construct appropriate matrices U, V (using $\vec{v}_i$'s as rows and columns) and a matrix Λ with the eigenvalues λ_i as components such that

$$A = U \Lambda V$$

.

Solution: First, we try to create a matrix operation that results in $\vec{v}_1 \vec{v}_1{}^T + \vec{v}_2 \vec{v}_2{}^T + \cdots + \vec{v}_n \vec{v}_n{}^T$ with matrices U and V. Note that $\vec{v}_i \vec{v}_i{}^T$ has to be a matrix, so $\vec{v}_i$ has to be a column vector. Therefore (remember the homework problem on "outer products" we had?),

$$\begin{bmatrix} \vec{v}_1 & \cdots & \vec{v}_n \end{bmatrix} \begin{bmatrix} \vec{v}_1{}^T \\ \vdots \\ \vec{v}_n{}^T \end{bmatrix} = \vec{v}_1 \vec{v}_1{}^T + \vec{v}_2 \vec{v}_2{}^T + \cdots + \vec{v}_n \vec{v}_n{}^T$$

We can let

$$U = \begin{bmatrix} \vec{v}_1 & \cdots & \vec{v}_n \end{bmatrix}, V = U^T$$

.

Now to add the scalar λ_is for each $\vec{v}_i \vec{v}_i{}^T$ pair, we can simply scale one set of vectors:

$$\begin{bmatrix} \lambda_1 \vec{v}_1 & \cdots & \lambda_n \vec{v}_n \end{bmatrix} \begin{bmatrix} \vec{v}_1{}^T \\ \vdots \\ \vec{v}_n{}^T \end{bmatrix} = \lambda_1 \vec{v}_1 \vec{v}_1{}^T + \lambda_2 \vec{v}_2 \vec{v}_2{}^T + \cdots + \lambda_n \vec{v}_n \vec{v}_n{}^T$$

Notice that we can scale the columns of U by λ_i by post-multiplying (multiplying on the right) by a diagonal matrix[1]:

$$\begin{bmatrix} \lambda_1 \vec{v}_1 & \cdots & \lambda_n \vec{v}_n \end{bmatrix} = \begin{bmatrix} \vec{v}_1 & \cdots & \vec{v}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Therefore, let

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

And we have

$$A = \lambda_1 \vec{v}_1 \vec{v}_1^{\,T} + \lambda_2 \vec{v}_2 \vec{v}_2^{\,T} + \cdots + \lambda_n \vec{v}_n \vec{v}_n^{\,T} = U \Lambda V$$

(b) Use the IPython notebook `prob12.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?

Solution:

The shape function told us that the image is a 400 by 400 matrix. We can therefore expect that we could have as many as 400 eigenvalues, and indeed look at the eig_vals variable in the notebook to see that all 400 of them are not zero. This tells us that to fully understand the action of the matrix, we would need to know about all 400 of the eigenspaces. So, we would need 400 eigenvectors to fully understand them.

At least, this is true without using deeper tricks. In reality, you might expect that the fact that this particular matrix/image is symmetric should allow us to exploit that symmetry somehow to reduce the amount of information required to perfectly capture the matrix. Indeed this is true.

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

Solution: Solution in `sol12.ipynb`.

Some of you might have noticed that there was a bug in the provided code. We had forgotten to take the absolute value of the eigenvalues before sorting. If you fixed that bug yourself, kudos to you! If you didn't fix that bug, you would see these weird diagonal lines the reconstructed images and a washed-out look that reduced the contrast. With the large negative eigenvalues intact, that effect does not happen.

Don't take points off if you didn't fix this bug. Give yourself a smiley face if you did.

(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?

Solution: Solution in `sol12.ipynb`.

The question of the lowest value of k is a bit subjective and it is fine whatever answer you gave for it. Personally, I think that the image starts looking qualitatively different somewhere around 15

[1] This has to be the case since λ_i times the ith standard unit vector $\vec{e}_i$ will have λ_i times the ith column of the matrix U as a response to the linear operation defined by U. It is because matrix multiplication on the right deals with columns that this makes sense.

eigenvectors. Certainly below 7, it looks very different. The "resolution" seems to be dropping as we keep fewer and fewer eigenvalues and eigenvectors.

Look at the plot of the eigenvalues included in the solutions notebook. You will see how they fast they become small.

This effect, of quality reducing as we keep less information, is something that all of you have experienced while using things like jpeg compression. We hope that seeing this example gives you some idea of why it could be possible to do such "lossy compression" in real-world applications. Later in the 16AB sequence, we will be learning more about why this works.

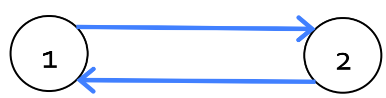
Along with lossless compression and error-correcting codes, lossy compression is one of the major advances that makes the modern age of multimedia possible. So, give a silent shoutout to eigenconcepts when you next watch a video online.

6. Counting the paths of a Random Surfer

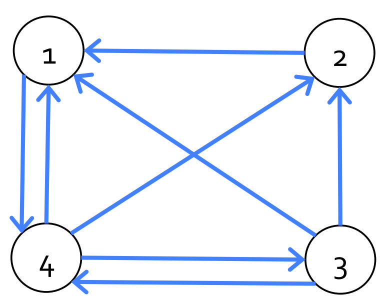
In class, we discussed the behavior of a random web-surfer who jumps from webpage to webpage. We would like to know how many possible paths there are for a random surfer to get from a page to another page. To do this, we represent the webpages as a graph. If page 1 has a link to page 2, we have a directed 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 a particular webpage i to webpage j.

This path counting aspect 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.

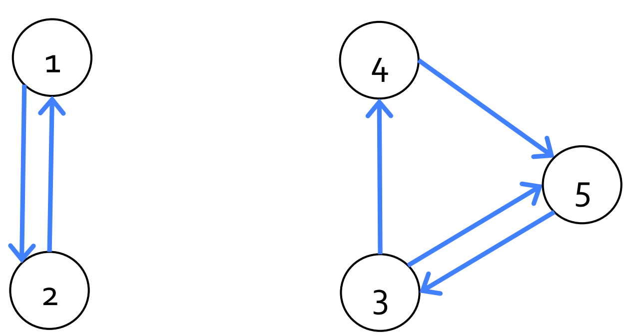
Consider the following graphs.



Graph A



Graph B



Graph C

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

Solution: The adjacency matrix only has 0s and 1s, representing whether a connection between two nodes exists or not. Therefore, it can be thought of as a way to represent the connectivity of the graph. Let A be the adjacency matrix for Graph A. Then,

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

.

(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?

Solution: We take the n^{th} power of the adjacency matrix to determine how many n-hops paths exist between the pages.

$$A^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Therefore, there is 1 one-hop path between webpage-1 and webpage-2 (which can be checked trivially).

$$A^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

There are no two-hop paths between webpage-1 and webpage-2! This matches the structure of the graph since two hops will always get the websurfer back to the page they started from.

Why does this work? Let's look at what the A^2 matrix multiplication does:

The first element (describing the path to get to and from node 1 in two hops) is (number of paths from node-1 to node-1)^2+(number of paths from node-2 to node-1)(number of paths from node-1 to node-2). This is $0+(1)(1)=1$. The result is therefore the sum of any self-loops and the number of paths going to node-2 and back. A similar formula applies for the n^{th} powers.

This is because the concatenation of two paths is a valid path if one ends where the other begins. So the number of n-hop paths from i to j must in fact be the sum, over all intermediate pages k, of the number of ℓ-hop paths from i to k times the number of $(n-\ell)$-hop paths from k to j. This is precisely what matrix multiplication does.

$$A^3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

There is 1 three-hop path between webpage-1 and webpage-2. Note that $A^3 = A$.

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

Solution:

To determine the importance score of the two pages, we need to find the appropriate eigenvector of the transition matrix. In this case, we are trying to determine the proportion of people who would be on a given page at steady state. Therefore, we use a transition matrix that deals with probabilities, instead of the connectivity (adjacency) matrix.

The transition matrix of graph A:

$$A_P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

To determine the eigenvalues of this matrix:

$$\det\left(\begin{bmatrix} -\lambda & 1 \\ 1 & -\lambda \end{bmatrix}\right) = \lambda^2 - 1 = 0$$

$\lambda = 1, -1$. The steady state vector is the eigenvector that corresponds to $\lambda = 1$. To find the eigenvector,

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

The sum of the values of the vector should equal 1, so our conditions are:

$$v_1 + v_2 = 1$$
$$v_1 = v_2$$

The importance score vector is $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$ and each webpage has an importance score of 0.5.

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

Solution: Let B be the adjacency matrix for Graph B. Then,

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

(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?

Solution: Using the same procedure as part (b),

$$B^2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

There is 1 two-hop path from webpage-1 to webpage-3. Let's look at what the B^2 matrix multiplication does:

The B_{31} element (describing the path to get from webpage-1 to webpage-3) is

(paths from node-1 to node-3)(paths from node-1 to node-1)+(paths from node-2 to node-3)(paths from node-1 to node-2)+(paths from node-3 to node-3)(paths from node-1 to node-3)+(paths from node-4 to node-3)(paths from node-1 to node-4)+(paths from node-5 to node-3)(paths from node-1 to node-5).

This is $0+0+0+1=1$. A similar formula applies for the n^{th} powers.

For three hop paths,

$$B^3 = \begin{bmatrix} 1 & 0 & 2 & 2 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 & 3 \\ 1 & 1 & 2 & 2 \\ 0 & 1 & 1 & 2 \\ 2 & 0 & 3 & 2 \end{bmatrix}$$

There is 1 three-hop path from webpage-1 to webpage-2.

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

Solution: To determine the importance scores, we need to create the transition matrix B_P first.

$$B_P = \begin{bmatrix} 0 & 1 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{3} \\ 1 & 0 & \frac{1}{3} & 0 \end{bmatrix}$$

The eigenvector associated with eigenvalue 1 is $\begin{bmatrix} -0.61 & -0.31 & -0.23 & -0.69 \end{bmatrix}^T$ (found using IPython). Scaling it appropriately so the elements add to 1, we get $\begin{bmatrix} \frac{1}{3} & \frac{1}{6} & \frac{1}{8} & \frac{3}{8} \end{bmatrix}^T$ These are the importance scores for the pages.

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

Solution: Let C be the adjacency matrix for Graph C. Then,

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

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

Solution: There are no paths from webpage-1 to webpage-3 (and no n-hop paths either).

(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?

Solution: The transition matrix for graph C is

$$C_P = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 & 0 \end{bmatrix}$$

The eigenvalues of this graph are $\lambda = 1, 1, -1, -\frac{1}{2} + -\frac{i}{2}, -\frac{1}{2} - -\frac{i}{2}$ (found using IPython). The eigenvectors associated with $\lambda = 1$ are $\begin{bmatrix} 0 & 0 & 0.4 & 0.2 & 0.4 \end{bmatrix}^T$ and $\begin{bmatrix} 0.5 & 0.5 & 0 & 0 & 0 \end{bmatrix}^T$. Why are there two eigenvectors? The first eigenvector describes the importance scores of the last three webpages, and the second vector describes the importance scores of the first two webpages. This makes sense since there are essentially “two internets", or two disjoint set of webpages. Surfers cannot transition between the two, so you cannot assign importance scores to webpage-1 and webpage-2 relative to the rest.

Assuming that each set of importance scores needs to add to 1, the first assigns importance scores of 0.4, 0.2, 0.4 to webpage-3 , webpage-4, and webpage-5, respectively. The second assigns importance scores of 0.5 to both webpage-1 and webpage-2.

7. 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 $i = 1,2,\dots,N$, 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.

$$s_i = \sum_{j}^{N} q_{ij} r_j, \tag{39}$$

where q_{ij} represents the number of times team i has beaten team j divided by the number [2] 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 their relationship as a system of equations

$$\vec{s} = Q\vec{r}, \tag{40}$$

where $Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ q_{N1} & q_{N2} & \cdots & q_{NN} \end{bmatrix}$ 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?

Solution:

$$\begin{bmatrix} 0 & \frac{1}{5} & \frac{2}{5} \\ \frac{2}{6} & 0 & 0 \\ 0 & \frac{3}{5} & 0 \end{bmatrix} \tag{41}$$

(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.

Solution:

With our assumption we have $\vec{s} = \lambda\vec{r}$, and thus $\lambda\vec{r} = Q\vec{r}$.

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

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

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,n, and λ

Solution: $\lambda^n c\vec{v}$

This is because $Q^n c\vec{v} = cQ^n\vec{v} = cQ^{n-1}\lambda\vec{v} = \lambda cQ^{n-1}\vec{v} = \cdots = \lambda^{n-1}cQ\vec{v} = \lambda^n c\vec{v}$.

(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.

Solution:

First we distribute Q to get

$$\sum_{i=1}^{m} Q^n c_i \vec{v}_i. \tag{42}$$

From the previous part we know that we can express each term in the summation with $\lambda_i^n c_i\vec{v}_i$, and thus we get

$$\sum_{i=1}^{m} \lambda_i^n c_i \vec{v}_i. \tag{43}$$

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

$$\lim_{n\to\infty} \frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^{m} c_i\vec{v}_i) = c_1\vec{v}_1 \tag{44}$$

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 $|w| < 1$, $\lim_{n\to\infty} w^n = 0$.

Solution:

From the previous part we can conclude that

$$\frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^{m} c_i\vec{v}_i) = \frac{1}{\lambda_1^n}\sum_{i=1}^{m} \lambda_i^n c_i\vec{v}_i, \tag{45}$$

which may be rewritten as

$$c_1\vec{v}_1 + \sum_{i=2}^{m}(\frac{\lambda_i}{\lambda_1})^n c_i\vec{v}_i, \tag{46}$$

where $|\frac{\lambda_i}{\lambda_1}| < 1$ for $i = 2, \ldots, m$, therefore $\lim_{n\to\infty}(\frac{\lambda_i}{\lambda_1})^n = 0$ and

$$\lim_{n\to\infty} \frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^{m} c_i\vec{v}_i) = c_1\vec{v}_1 \tag{47}$$

(f) Now further assuming that λ_1 is positive show

$$\lim_{n\to\infty} \frac{Q^n(\sum_{i=1}^{m} c_i\vec{v}_i)}{||Q^n(\sum_{i=1}^{m} c_i\vec{v}_i)||} = \frac{c_1\vec{v}_1}{||c_1\vec{v}_1||} \tag{48}$$

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 $\lim_{n\to\infty} \frac{\vec{a}_n}{\alpha_n} = \frac{\lim_{n\to\infty} \vec{a}_n}{\lim_{n\to\infty} \alpha_n}$.

Solution: First we use the hint and write the expression

$$\lim_{n\to\infty} \frac{\frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^m c_i\vec{v}_i)}{\frac{1}{\lambda_1^n}||Q^n(\sum_{i=1}^m c_i\vec{v}_i)||}. \tag{49}$$

Using that λ_1 is positive we get

$$\lim_{n\to\infty} \frac{\frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^m c_i\vec{v}_i)}{||\frac{1}{\lambda_1^n}Q^n(\sum_{i=1}^m c_i\vec{v}_i)||}, \tag{50}$$

Since the denominator does not converge to zero we get

$$\lim_{n\to\infty} \frac{\frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^m c_i\vec{v}_i)}{||\frac{1}{\lambda_1^n}Q^n(\sum_{i=1}^m c_i\vec{v}_i)||} = \frac{\lim_{n\to\infty} \frac{1}{\lambda_1^n} Q^n(\sum_{i=1}^m c_i\vec{v}_i)}{||\lim_{n\to\infty} \frac{1}{\lambda_1^n}Q^n(\sum_{i=1}^m c_i\vec{v}_i)||}. \tag{51}$$

Finally, using our result from the previous part we obtain

$$\lim_{n\to\infty} \frac{Q^n(\sum_{i=1}^m c_i\vec{v}_i)}{||Q^n(\sum_{i=1}^m c_i\vec{v}_i)||} = \frac{c_1\vec{v}_1}{||c_1\vec{v}_1||}. \tag{52}$$

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 [3]

$$\lim_{n\to\infty} \frac{Q^n\vec{b}}{||Q^n\vec{b}||} = \frac{c_1\vec{v}_D}{||c_1\vec{v}_D||}. \tag{53}$$

This is the idea behind the power iteration method, which is a method for finding the unique dominant eigenvector (up to scale) 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.

Solution:

Oregon, Alabama, Arizona, Mississippi, UCLA, LSU, USC.

Here is an example of the code that could have been entered for the power iteration method

```
b=np.dot(Q,b)/np.linalg.norm(np.dot(Q,b))
```

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

8. The Dynamics of Romeo and Juliet's Love Affair

In this problem we study a discrete-time model of the dynamics of Romeo and Juliet's love affair—adapted from Steven H. Strogatz's original paper, *Love Affairs and Differential Equations*, Mathematics Magazine, 61(1), p.35, 1988, which described a continuous-time model.

Let $R[n]$ denote Romeo's feelings about Juliet on day n, and let $J[n]$ quantify Juliet's feelings about Romeo on day n. If $R[n] > 0$, it means that Romeo loves Juliet and inclines toward her, whereas if $R[n] < 0$, it means that Romeo is resentful of her and inclines away from her. A similar interpretation holds for $J[n]$, which represents Juliet's feelings about Romeo.

A larger $|R[n]|$ represents a more intense feeling of love (if $R[n] > 0$) or resentment (if $R[n] < 0$). If $R[n] = 0$, it means that Romeo has neutral feelings toward Juliet on day n. Similar interpretations hold for larger $|J[n]|$ and the case of $J[n] = 0$.

We model the dynamics of Romeo and Juliet's relationship using the following coupled system of linear evolutionary equations:

$$R[n+1] = a\,R[n] + b\,J[n], \quad n = 0, 1, 2, \ldots \tag{54}$$

and

$$J[n+1] = c\,R[n] + d\,J[n], \quad n = 0, 1, 2, \ldots, \tag{55}$$

which we can rewrite as

$$\vec{s}[n+1] = \mathbf{A}\,\vec{s}[n], \tag{56}$$

where

$$\vec{s}[n] = \begin{bmatrix} R[n] \\ J[n] \end{bmatrix}$$

denotes the state vector, and

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

the state-transition matrix, for our dynamic system model.

The parameters a and d capture the linear fashion in which Romeo and Juliet respond to their own feelings, respectively, about the other person. It's reasonable to assume that $a, d > 0$, to avoid scenarios of fluctuating day-to-day mood swings. Within this positive range, if $0 < a < 1$, then the effect of Romeo's own feelings about Juliet tend to fizzle away with time (in the absence of influence from Juliet to the contrary), whereas if $a > 1$, Romeo's feelings about Juliet intensify with time (in the absence of influence from Juliet to the contrary). A similar interpretation holds if $0 < d < 1$ and $d > 1$.

The parameters b and c capture the linear fashion in which the other person's feelings influences $R[n]$ and $J[n]$, respectively. These parameters may or may not be positive. If $b > 0$, it means that the more Juliet shows affection for Romeo, the more he loves her and inclines toward her. If $b < 0$, it means that the more Juliet shows affection for Romeo, the more resentful he feels and the more he inclines away from her. A similar interpretation holds for the parameter c.

All in all, each of Romeo and Juliet has four romantic styles, which makes for a combined total of sixteen possible dynamic scenarios. And the fate of their interactions depends on the romantic style each of them exhibits, the initial state, and the values of values of the entries in the state-transition matrix $\mathbf{A}$. In this problem, we'll explore a subset of the possibilities.

(a) Consider the case where $a+b=c+d$ in the state-transition matrix

$$\mathbf{A}=\begin{bmatrix} a & b \\ c & d \end{bmatrix}.$$

i. Show that

$$\vec{v}_1=\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

is an eigenvector of $\mathbf{A}$, and determine its corresponding eigenvalue λ_1. Determine, also, the other eigenpair $(\lambda_2, \vec{v}_2)$. Your expressions for λ_1, λ_2, and $\vec{v}_2$ must be in terms of one or more of the parameters a, b, c, and d.

Solution:

$$\begin{aligned}
\mathbf{A}\begin{bmatrix} 1 \\ 1 \end{bmatrix} &= \begin{bmatrix} a+b \\ c+d \end{bmatrix} \\
&= (a+b)\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
&= (c+d)\begin{bmatrix} 1 \\ 1 \end{bmatrix}
\end{aligned}$$

Let $\mu=a+b=c+d$. Then it's clear that $\begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathsf{T}}$ is an eigenvector of $\mathbf{A}$ corresponding to the eigenvalue μ. So the following is an eigenpair of $\mathbf{A}$:

$$\left(\lambda_1=a+b=c+d, \vec{v}_1=\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right).$$

To determine the other eigenpair $(\lambda_2, \vec{v}_2)$, we determine the other eigenvalue λ_2 first. We can do this in one of two ways:

Method I: Use the fact that $\mathrm{tr}\mathbf{A}=\lambda_1+\lambda_2$. We therefore have

$$\begin{aligned}
a+d &= \lambda_1+\lambda_2 \\
&= a+b+\lambda_2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\lambda_2 &= a+d-\lambda_1 \\
&= a+d-(a+b) \\
&= d-b.
\end{aligned}$$

If we use the expression $\lambda_1=c+d$, then an identical approach yields $\lambda_2=a-c$.

Method II: An alternative approach is to determine the second eigenvalue λ_2 by solving the characteristic polynomial

$$\begin{aligned}
\det(\lambda\mathbf{I}-\mathbf{A}) &= \det\left(\begin{bmatrix} \lambda-a & -b \\ -c & \lambda-d \end{bmatrix}\right) \\
&= (\lambda-a)(\lambda-d)-bc \\
&= \lambda^2-(a+d)\lambda-bc \\
&= 0.
\end{aligned}$$

We know from theory of quadratic polynomials that the sum of the roots equals the negative of the coefficient of the linear term λ. So, $\lambda_1 + \lambda_2 = a + d$. Notice that for a 2×2 matrix, the coefficient of λ is $-\text{tr}\mathbf{A}$. And we can now use the same steps of Method I from here on.

Once we have the second eigenvalue, we use it to build the matrix $\lambda_2 \mathbf{I} - \mathbf{A}$. However, we do this in a smart way. We use the expression $\lambda_1 = a - c$ for the first row, and $\lambda_1 = d - b$ for the second row. That is,

$$\begin{aligned} \lambda_2 \mathbf{I} - \mathbf{A} &= \begin{bmatrix} (a-c)-a & -b \\ -c & (d-b)-d \end{bmatrix} \\ &= \begin{bmatrix} -c & -b \\ -c & -b \end{bmatrix}. \end{aligned}$$

Clearly, $\lambda_2 \mathbf{I} - \mathbf{A}$ has linearly dependent columns, and the vector

$$\vec{v}_2 = \begin{bmatrix} b \\ -c \end{bmatrix}$$

lies in its nullspace. So, we have our second eigenpair:

$$\left(\lambda_2 = a - c = d - b, \vec{v}_2 = \begin{bmatrix} b \\ -c \end{bmatrix} \right).$$

Observation: You should note that any matrix whose row sums are a constant, say μ, must have $(\mu, \vec{1})$ as an eigenpair, where $\vec{1}$ is the all-ones vector of appropriate size.

ii. Consider the following state-transition matrix:

$$\mathbf{A} = \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix}.$$

i. Determine the eigenpairs for this system.

Solution: Notice that in this matrix, $a = d = 0.75$ and $b = c = 0.25$. So $\mu = a - c = d - b = 0.5$. Clearly, this is a row-stochastic matrix—each of its rows sums to 1. From the results of part (a)(I), we know that the eigenpairs of this matrix are

$$\left(\lambda_1 = 1, \vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right) \qquad \text{and} \qquad \left(\lambda_2 = 0.5, \vec{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right).$$

Observation: Notice that the eigenvectors $\vec{v}_1$ and $\vec{v}_2$ are orthogonal. This is not a coincidence. It turns out that the eigenvectors of a symmetric matrix are mutually orthogonal.

ii. Determine all the *fixed points* of the system. That is, find the set of points such that if Romeo and Juliet start at, or enter, any of those points, they'll stay in place forever: $\{\vec{s}_* | \mathbf{A}\vec{s}_* = \vec{s}_*\}$. Show these points on a diagram where the x- and y- axes are $R[n]$ and $J[n]$.

Solution: Any point along vector $\vec{s}_* = v_1 = \vec{1}$ is a fixed point, because $\vec{v}_1 = \vec{1}$ corresponds to the eigenvalue $\lambda_1 = 1$.

iii. Determine representative points along the state trajectory $\vec{s}[n]$, $n = 0, 1, 2, \ldots$, if Romeo and Juliet start from the initial state

$$\vec{s}[0] = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Solution: The general solution is given by

$$\begin{aligned}\vec{s}[n] &= \alpha_1\,\lambda_1^n\,\vec{v}_1 + \alpha_2\,\lambda_2^n\,\vec{v}_2 \\ &= \alpha_1 1^n \begin{bmatrix}1\\1\end{bmatrix} + \alpha_1 0.5^n \begin{bmatrix}1\\-1\end{bmatrix}.\end{aligned}$$

Since $\vec{v}_1 \perp \vec{v}_2$, and since $\vec{s}[0] = \vec{v}_2$, we know that $\alpha_1 = 0$ and $\alpha_2 = 1$. Therefore,

$$\vec{s}[n] = 0.5^n \begin{bmatrix}1\\-1\end{bmatrix}.$$

Since 0.5^n decays to zero as $n \to \infty$, the state trajectory stays along the second eigenvector and decays to the origin:

$$\lim_{n\to\infty}(R[n], J[n]) = (0,0).$$

In particular, the state vector obeys the following trajectory:

$$\begin{bmatrix}R[n]\\J[n]\end{bmatrix} = \begin{bmatrix}\left(\frac{1}{2}\right)^n\\ -\left(\frac{1}{2}\right)^n\end{bmatrix}, \quad n = 0,1,2,\ldots$$

This means that, ultimately, Romeo and Juliet will become neutral to each other.

iv. Suppose the initial state is $\vec{s}[0] = [3\ 5]^{\mathsf{T}}$. Determine a reasonably simple expression for the state vector $\vec{s}[n]$. Find the limiting state vector

$$\lim_{n\to\infty} \vec{s}[n].$$

Solution: We must express the initial state vector as a linear combination of the eigenvectors. That is, we must solve the system of linear equations

$$\begin{aligned}\begin{bmatrix}\vec{v}_1 & \vec{v}_2\end{bmatrix}\vec{\alpha} &= \vec{s}[0] \\ \begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}\alpha_1\\\alpha_2\end{bmatrix} &= \begin{bmatrix}3\\5\end{bmatrix}.\end{aligned}$$

It's straightforward to find the solution:

$$\vec{\alpha} = \begin{bmatrix}\alpha_1\\\alpha_2\end{bmatrix} = \begin{bmatrix}4\\-1\end{bmatrix}.$$

Therefore, the state vector is given by

$$\begin{aligned}\vec{s}[n] &= \alpha_1\,\lambda_1^n\,\vec{v}_1 + \alpha_2\,\lambda_2^n\,\vec{v}_2 \\ &= 4\begin{bmatrix}1\\1\end{bmatrix} - \left(\frac{1}{2}\right)^n \begin{bmatrix}1\\-1\end{bmatrix} \\ &= \begin{bmatrix}4 - \left(\frac{1}{2}\right)^n\\ 4 + \left(\frac{1}{2}\right)^n\end{bmatrix}\end{aligned}$$

Clearly,

$$\lim_{n\to\infty}\vec{s}[n] = \begin{bmatrix}4\\4\end{bmatrix}.$$

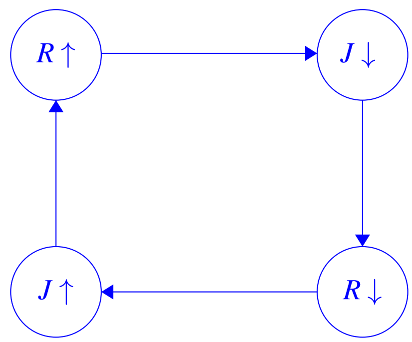
(b) Consider the setup in which

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

In this scenario, if Juliet shows affection toward Romeo, Romeo's love for her increases, and he inclines toward her. The more intensely Romeo inclines toward her, the more Juliet distances herself. The more Juliet withdraws, the more Romeo is discouraged and retreats into his cave. But the more Romeo inclines away, the more Juliet finds him attractive and the more intensely she conveys her affection toward him. Juliet's increasing warmth increases Romeo's interest in her, which prompts him to incline toward her—again!

Predict the outcome of this scenario before you write down a single equation.

Solution: We expect a never-ending cycle—an oscillation. The following diagram shows a qualitative picture of what happens.



Beginning with the top-left node, we see that Romeo's affection increases. As a result, Juliet retreats, as depicted by the node on the top-right. In turn, this causes Romeo to lose hope and retreat, as shown in the bottom-right node. When Romeo pulls away, Juliet finds him mystically attractive and gravitates toward him, as shown by the bottom-left node. This causes Romeo to turn toward Juliet, which takes us back to the top-left node again, for yet another cycle.

Then determine a complete solution $\vec{s}[n]$ in the simplest of terms, assuming an initial state given by $\vec{s}[0] = [1\ 0]^{\mathsf{T}}$. As part of this, you must determine the eigenvalues and eigenvectors of the $\mathbf{A}$.

Solution: The eigenvalues are the roots of the equation

$$\begin{aligned} \det(\lambda \mathbf{I} - \mathbf{A}) &= \det \begin{bmatrix} \lambda & -1 \\ 1 & \lambda \end{bmatrix} \\ &= \lambda^2 + 1 = 0. \end{aligned}$$

So, $\lambda_1 = i$ and $\lambda_2 = -i$. Constructing the matrices $\lambda_1 \mathbf{I} - \mathbf{A}$ and $\lambda_2 \mathbf{I} - \mathbf{A}$, we find the corresponding eigenvectors by inspection:

$$\lambda_1 \mathbf{I} - \mathbf{A} = \det \begin{bmatrix} i & -1 \\ 1 & i \end{bmatrix} \quad \text{so} \quad \vec{v}_1 = \begin{bmatrix} 1 \\ i \end{bmatrix}$$

and

$$\lambda_2 \mathbf{I} - \mathbf{A} = \det \begin{bmatrix} -i & -1 \\ 1 & -i \end{bmatrix} \quad \text{so} \quad \vec{v}_2 = \begin{bmatrix} 1 \\ -i \end{bmatrix}.$$

The matrix $\mathbf{A}$ has complex-valued eigenvalues and eigenvectors. Specifically, it has purely imaginary eigenvalues. This is not a coincidence. It turns out that if a matrix $\mathbf{A}$ has odd symmetry—that is, if $\mathbf{A}^{\mathsf{T}} = -\mathbf{A}$—then its eigenvalues are purely imaginary.

Before we determine the general solution $\vec{s}[n]$, we must decompose the initial-state vector in terms of the two eigenvectors. The equation is

$$\underbrace{\begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}}_{[\vec{v}_1 \ \ \vec{v}_2]} \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix}}_{\alpha} = \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\vec{s}[0]},$$

which yields the coefficient vector

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}.$$

The general solution is given by

$$\begin{aligned} \vec{s}[n] &= \alpha_1 \lambda_1^n \vec{v}_1 + \alpha_2 \lambda_2^n \vec{v}_2 \\ &= \frac{1}{2} i^n \begin{bmatrix} 1 \\ i \end{bmatrix} + \frac{1}{2} (-i)^n \begin{bmatrix} 1 \\ -i \end{bmatrix}. \end{aligned}$$

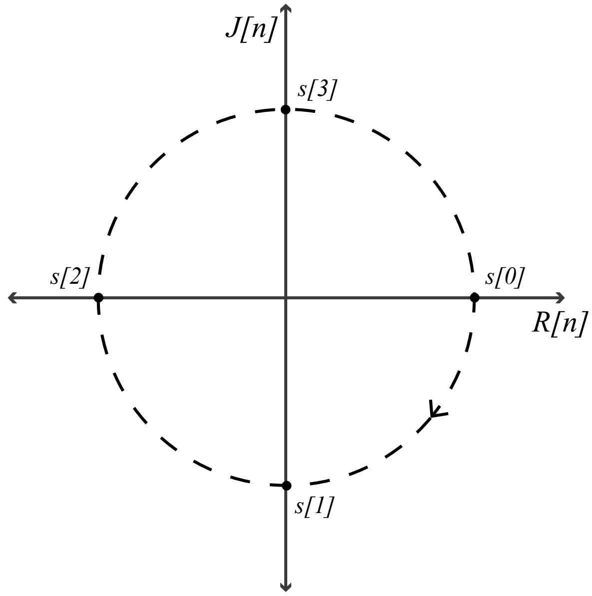
Since the two terms on the right-hand side are complex conjugates of one another, we have

$$\begin{aligned} \vec{s}[n] &= 2\,\mathsf{Re}\left(\frac{1}{2} i^n \begin{bmatrix} 1 \\ i \end{bmatrix} \right) \\ &= \mathsf{Re} \begin{bmatrix} i^n \\ i^{n+1} \end{bmatrix} \end{aligned}$$

$$\vec{s}[n] = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \text{if } n \geq 0, \text{and } n \mod 4 = 0 \\ \begin{bmatrix} 0 \\ -1 \end{bmatrix} & \text{if } n \geq 0, \text{and } n \mod 4 = 1 \\ \begin{bmatrix} -1 \\ 0 \end{bmatrix} & \text{if } n \geq 0, \text{and } n \mod 4 = 2 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix} & \text{if } n \geq 0, \text{and } n \mod 4 = 3 \end{cases}$$

Plot (by hand, or otherwise without the assistance of any scientific computing software package), on a two-dimensional plane (called a *phase plane*)—where the horizontal axis denotes $R[n]$ and the vertical axis denotes $J[n]$—representative points along the trajectory of the state vector $\vec{s}[n]$, starting from the initial state given in this part. Describe, in plain words, what Romeo and Juliet are doing in this scenario. In other words, what does their state trajectory look like? Determine $\|\vec{s}[n]\|^2$ for all $n = 0, 1, 2, \ldots$ to corroborate your description of the state trajectory.

Solution: Romeo and Juliet are going around in a clockwise circle. Note that $\|\vec{s}[n]\|^2 = 1$ for all $n = 0, 1, 2, 3, \ldots$.


J[n]
s[3]
s[2]
s[0]
R[n]
s[1]