EECS 16A Designing Information Devices and Systems I
Spring 2020 Homework 5

This homework is due February 28, 2020, at 23:59.
Self-grades are due March 3, 2020, at 23:59.

Submission Format
Your homework submission should consist of **one** file.

- `hw5.pdf`: A single PDF file that contains all of your answers (any handwritten answers should be scanned) as well as your IPython notebook saved as a PDF.

  If you do not attach a PDF "printout" of your IPython notebook, you will not receive credit for problems that involve coding. Make sure that your results and your plots are visible. Assign the IPython printout to the correct problem(s) on Gradescope.

- Practice problems will not be graded. However, they do provide more practice and we encourage you to try them to practice for the midterm.

- We encourage you to turn in self-grades for this HW before the midterm, since looking at the solutions earlier will help you study for the midterm.

Submit the file to the appropriate assignment on Gradescope.

1. Mechanical Determinants

For each of the following matrices, compute their determinant and state whether they are invertible.

(a) $\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}$.

Solution:

We can use the form of a 2×2 determinant from lecture:

$$\det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

Therefore,

$$\det\left(\begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix}\right) = 2 \cdot 3 - 0 = 6$$

Since the determinant is not 0, the matrix is invertible.

(b) $\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$.

Solution:

$$\det\left(\begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}\right) = 2 \cdot 3 - 1 \cdot 0 = 6$$

Since the determinant is not 0, the matrix is invertible.

(c) $\begin{bmatrix} 6 & 9 \\ 4 & 6 \end{bmatrix}$.

Solution:

$$\det\left(\begin{bmatrix} 6 & 9 \\ 4 & 6 \end{bmatrix}\right) = 6\cdot 6 - 9\cdot 4 = 0$$

Since the determinant is 0, the matrix is non-invertible.

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

Learning Goal: *Eigenvalues and eigenvectors of state transition matrices tend to reveal useful information about the dynamical systems they model. This problem serves as an example of extracting useful information through analysis of the eigenvalues of the state transition matrix of a dynamical system.*

In this problem, we will 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 describes a continuous-time model.

Let $R[n]$ denote Romeo's feelings about Juliet on day n, and let $J[n]$ denote Juliet's feelings about Romeo on day n. The sign of $R[n]$ (or $J[n]$) indicates like or dislike. For example, if $R[n] > 0$, it means Romeo likes Juliet. On the other hand, $R[n] < 0$ indicates that Romeo dislikes Juliet. $R[n] = 0$ indicates that Romeo has a neutral stance towards Juliet.

The magnitude (i.e. absolute value) of $R[n]$ (or $J[n]$) represents the intensity of that feeling. For example, a larger $|R[n]|$ means that Romeo has a stronger emotion towards Juliet (love if $R[n] > 0$ or hatred if $R[n] < 0$). Similar interpretations hold for $J[n]$.

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

$$R[n+1] = aR[n] + bJ[n], \quad n = 0, 1, 2, \ldots$$

and

$$J[n+1] = cR[n] + dJ[n], \quad n = 0, 1, 2, \ldots,$$

which we can rewrite as

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

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 selection of the parameters a, b, c, d results in different dynamic scenarios. The fate of Romeo and Juliet's relationship depends on these model parameters (i.e. a, b, c, d) in the state transition matrix and the initial state ($\vec{s}[0]$). In this problem, we'll explore some of these possibilities.

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

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

Show that

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

is an eigenvector of $\mathbf{A}$, and determine its corresponding eigenvalue λ_1. Show that

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

is an eigenvector of $\mathbf{A}$, and determine its corresponding eigenvalue λ_2. Now, express the first and second eigenvalues and their eigenspaces in terms of the parameters $a, b, c,$ and d.

Hint: Consider $\mathbf{A}\vec{v}_1$. Is it equal to a scalar multiple of $\vec{v}_1$? Similarly for $\vec{v}_2$.

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 $\lambda_1 = a+b = c+d$. Then you can plug in to find that $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$ is an eigenvector of $\mathbf{A}$ corresponding to the eigenvalue λ_1. The first eigenpair $\mathbf{A}$ is,

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

To determine the other eigenvalues and corresponding eigenvectors $(\lambda_2, \vec{v}_2)$, we use the hint that $\vec{v}_2 = \begin{bmatrix} b \\ -c \end{bmatrix}$. Note that by modifying the constraint $a+b = c+d$, we can also get $a-c = d-b$, which helps simplify the following:

$$\begin{aligned} \mathbf{A}\begin{bmatrix} b \\ -c \end{bmatrix} &= \begin{bmatrix} ab-bc \\ cb-dc \end{bmatrix} \\ &= \begin{bmatrix} b(a-c) \\ -c(d-b) \end{bmatrix} \\ &= (a-c)\begin{bmatrix} b \\ -c \end{bmatrix} \\ &= (d-b)\begin{bmatrix} b \\ -c \end{bmatrix} \end{aligned}$$

Therefore, we have our second eigenpair:

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

For parts (b) - (e), consider the following state-transition matrix:

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

(b) Determine the eigenvalues and corresponding eigenvectors (i.e. $\lambda_1, \vec{v}_1$ and $\lambda_2, \vec{v}_2$) for this system. Note that this matrix is a special case of the matrix explored in part (a), so you can use results from that part to help you.

Solution:

From the results of part (a), we know that the eigenvalues and eigenvectors of this matrix are

$$\left(\lambda_1 = a + b = 0.75 + 0.25 = 1, \vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

and

$$\left(\lambda_2 = a - c = 0.75 - 0.25 = 0.5, \vec{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right).$$

Note: If your choice of eigenvector $\vec{v}_1$ and $\vec{v}_2$ is a scaled version of the ones given in this solution, that is fine.

(c) Determine all of the *steady states* of the system. That is, find the set of points such that if Romeo and Juliet start at, or enter, any of those points, their states will stay in place forever: $\{\vec{s}_* \mid \mathbf{A}\vec{s}_* = \vec{s}_*\}$.

Solution: Any $\vec{s}_* \in \text{span}\{\vec{v}_1\}$, where $\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, is the eigenvector which corresponds to the steady state, because $\vec{v}_1$ corresponds to the eigenvalue $\lambda_1 = 1$.

(d) Suppose Romeo and Juliet start from an initial state $\vec{s}[0] \in \text{span}\left\{\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right\}$. What happens to their relationship over time? Specifically, what is $\vec{s}[n]$ as $n \to \infty$?

Solution:

We note that $\vec{s}[0] \in \text{span}\{\vec{v}_2\}$. Therefore,

$$\begin{aligned} \vec{s}[1] &= \mathbf{A}\vec{s}[0] \\ &= \alpha\lambda_2\vec{v}_2 \end{aligned}$$

where α is the scalar that expresses $\vec{s}[0]$ as a scaled version of $\vec{v}_2$.

If we continue to apply the state transition matrix, we will see that for this $\vec{s}[0]$,

$$\begin{aligned} \vec{s}[n] &= \mathbf{A}^n\vec{s}[0] \\ &= \alpha\lambda_2^n\vec{v}_2 \end{aligned}$$

In this case $\lambda_2 = 0.5$. This means that as $n \to \infty$, $\lambda_2^n \to 0$.

Therefore,

$$\begin{aligned} \vec{s}[n] &= \alpha\lambda_2^n\vec{v}_2 \\ &= \alpha \cdot 0 \cdot \vec{v}_2 \\ &= \vec{0} \end{aligned}$$

which means that

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

So, ultimately, Romeo and Juliet will become neutral to each other.

(e) Suppose the initial state is $\vec{s}[0] = \begin{bmatrix}3\\5\end{bmatrix}$. What happens to their relationship over time? Specifically, what is $\vec{s}[n]$ as $n \to \infty$?
Hint: Can you use what you learned about the eigenvectors of A (in parts c and d) to help you solve this problem? How can you represent the starting state?

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

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

You can row-reduce to find the solution:

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

Therefore, the starting state vector is given by

$$\vec{s}[0] = \alpha_1\vec{v}_1 + \alpha_2\vec{v}_2 = 4\vec{v}_1 - 1\vec{v}_2$$

If we apply the state transition matrix,

$$\begin{aligned}\vec{s}[1] &= \mathbf{A}\vec{s}[0]\\ &= \alpha_1\lambda_1\vec{v}_1 + \alpha_2\lambda_2\vec{v}_2\\ &= 4\lambda_1\vec{v}_1 - 1\lambda_2\vec{v}_2\end{aligned}$$

If we continue to apply the state transition matrix, we find:

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

Taking the limit as $n \to \infty$,

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

Romeo and Juliet's relationship converges to the state vector $\vec{s}[n] = \begin{bmatrix}4\\4\end{bmatrix}$.

Now suppose we have the following state-transition matrix:

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

Use this state-transition matrix for parts (f) - (h).

(f) Determine the eigenvalues and corresponding eigenvectors (i.e. $\lambda_1, \vec{v}_1$ and $\lambda_2, \vec{v}_2$) for this system. Note that this matrix is a special case of the matrix explored in part (a), so you can use results from that part to help you.

Solution: From the results of part (a), we know that the eigenvalues and corresponding eigenvectors of this matrix are

$$\left(\lambda_1 = a + b = 1 + 1 = 2, \vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

and

$$\left(\lambda_2 = a - c = 1 - 1 = 0, \vec{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right).$$

(g) Suppose Romeo and Juliet start from an initial state $\vec{s}[0] \in \text{span}\left\{\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right\}$. What happens to their relationship over time? Specifically, what is $\vec{s}[n]$ as $n \to \infty$?

Solution: The initial state $\vec{s}[0]$ lies in the span of the eigenvector $\vec{v}_2$, which has eigenvalue $\lambda_2 = 0$. Thus, $\vec{s}[1] = \vec{0}$. The state will remain at $\vec{0}$ for all subsequent time steps, i.e.

$$\vec{s}[n] = \vec{0}, n \geq 1$$

Therefore, Romeo and Juliet become neutral towards each other in the long run, i.e.

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

(h) Now suppose that Romeo and Juliet start from an initial state $\vec{s}[0] \in \text{span}\left\{\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right\}$. What happens to their relationship over time? Specifically, what is $\vec{s}[n]$ as $n \to \infty$? **Solution:**

We note that $\vec{s}[0] \in \text{span}\{\vec{v}_1\}$. Therefore,

$$\begin{aligned} \vec{s}[1] &= \mathbf{A}\vec{s}[0] \\ &= \alpha\lambda_1\vec{v}_1 \end{aligned}$$

where α is the scalar that expresses $\vec{s}[0]$ as a scaled version of $\vec{v}_1$.

If we continue to apply the state transition matrix, we will see that for this $\vec{s}[0]$,

$$\begin{aligned} \vec{s}[n] &= \mathbf{A}^n\vec{s}[0] \\ &= \alpha\lambda_1^n\vec{v}_1 \end{aligned}$$

In this problem, $\lambda_1 = 2$. Therefore,

$$\vec{s}[n] = \alpha 2^n \vec{v}_1$$

This means that as $n \to \infty$, $\lambda_1^n \to \infty$. Essentially, the elements of the state vector continue to double at each time step and grow without bound to either $+\infty$ or $-\infty$.
Therefore, what happens to Romeo and Juliet depends on $\vec{s}[0]$. If $\vec{s}[0]$ is in the first quadrant, Romeo and Juliet will become "infinitely" in love with each other. On the other hand, if $\vec{s}[0]$ is in the third quadrant, then Romeo and Juliet will have "infinite" hatred for each other.

Finally, we consider the case where we have the following state-transition matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$$

Use this state-transition matrix for parts (i) - (k).

(i) Determine the eigenvalues and corresponding eigenvectors (i.e. $\lambda_1, \vec{v}_1$ and $\lambda_2, \vec{v}_2$) for this system. Note that this matrix is a special case of the matrix explored in part (a), so you can use results from that part to help you.
Solution: From the results of part (a), we know that the eigenvalues and corresponding eigenvectors of this matrix are

$$\left(\lambda_1 = a + b = 1 - 2 = -1, \vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

and

$$\left(\lambda_2 = a - c = 1 - (-2) = 3, \vec{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}\right).$$

(j) Suppose Romeo and Juliet start from an initial state $\vec{s}[0] \in \text{span}\left\{\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right\}$. What happens to their relationship over time if $R[0] > 0$ and $J[0] < 0$? What about if $R[0] < 0$ and $J[0] > 0$? Specifically, what is $\vec{s}[n]$ as $n \to \infty$?
Solution: The initial state $\vec{s}[0]$ lies in the span of the eigenvector $\vec{v}_2$, which has eigenvalue $\lambda_2 = 3$. Using similar methods to the solutions in part (d) and part (h), we can see that (for a given scalar α):

$$\begin{aligned} \vec{s}[n] &= \mathbf{A}^n \vec{s}[0] \\ &= \alpha \lambda_2^n \vec{v}_2 \\ &= \alpha 3^n \vec{v}_2 \end{aligned}$$

There are two cases of long-term behavior.
Suppose, initially, that $R[0] > 0$ and $J[0] < 0$ (corresponding to $\alpha > 0$). Then as $n \to \infty$, $R[n] \to \infty$ and $J[n] \to -\infty$. Romeo will have "infinite" love for Juliet, while Juliet will have "infinite" hatred for Romeo.
Conversely, if initially $R[0] < 0$ and $J[0] > 0$ (corresponding to $\alpha < 0$), then as $n \to \infty$, $R[n] \to -\infty$ and $J[n] \to \infty$. Now Romeo would have "infinite" hatred for Juliet, while Juliet would have "infinite" love for Romeo.

(k) Now suppose that Romeo and Juliet start from an initial state $\vec{s}[0] \in \text{span}\left\{\begin{bmatrix}1\\1\end{bmatrix}\right\}$. What happens to their relationship over time? Specifically, what is $\vec{s}[n]$ as $n \to \infty$?

Solution: The initial state $\vec{s}[0]$ lies in the span of the eigenvector $\vec{v}_1$, which has eigenvalue $\lambda_1 = -1$. As with parts (d), (g), and (i), we can see that (for a given scalar α):

$$\begin{aligned}\vec{s}[n] &= \mathbf{A}^n \vec{s}[0] \\ &= \alpha \lambda_1^n \vec{v}_1 \\ &= \alpha(-1)^n \vec{v}_1\end{aligned}$$

The elements of the state vector continue to switch signs at each time step, while keeping the same magnitude.

Essentially, Romeo and Juliet maintain the same intensity (i.e. absolute value or magnitude) of feeling, but they keep changing their mind about whether that feeling is like or dislike at each time step. Note that $R[0]$ and $J[0]$ have the same sign, so they both either like each other or dislike each other at a given time step n.

3. Noisy Images

Learning Goal: *The third part of the imaging lab uses the eigenvalues of the masking matrix to understand which masks are better than others for image reconstruction in the presence of additive noise. This problem explores the underlying mathematics.*

In lab, we used a single pixel camera to capture many measurements of an image $\vec{i}$. A single scalar measurement s_i is captured using a mask $\vec{h}_i$ such that $s_i = \vec{h}_i^T \vec{i}$. Many measurements can be expressed as a matrix-vector multiplication of the masks with the image, where the masks lie along the rows of the matrix.

$$\begin{bmatrix}s_1\\ \vdots \\ s_N\end{bmatrix} = \begin{bmatrix}\vec{h}_1\\ \vdots \\ \vec{h}_N\end{bmatrix} \vec{i} \tag{1}$$

$$\vec{s} = \mathbf{H}\vec{i} \tag{2}$$

In the real world, noise, $\vec{w}$, creeps into our measurements, so instead,

$$\vec{s} = \mathbf{H}\vec{i} + \vec{w} \tag{3}$$

(a) Express $\vec{i}$ in terms of $\mathbf{H}$ (or its inverse), $\vec{s}$, and $\vec{w}$. Assume $\mathbf{H}$ is invertible. (*Hint:* Think about what you did in the imaging lab.)

Solution:

$$\vec{s} = \mathbf{H}\vec{i} + \vec{w} \tag{4}$$

$$\mathbf{H}^{-1}\vec{s} = \mathbf{H}^{-1}(\mathbf{H}\vec{i} + \vec{w}) \tag{5}$$

$$\mathbf{H}^{-1}\vec{s} = \mathbf{H}^{-1}\mathbf{H}\vec{i} + \mathbf{H}^{-1}\vec{w} \tag{6}$$

$$\mathbf{H}^{-1}\vec{s} = \vec{i} + \mathbf{H}^{-1}\vec{w} \tag{7}$$

$$\vec{i} = \mathbf{H}^{-1}\vec{s} - \mathbf{H}^{-1}\vec{w} \tag{8}$$

(b) Depending on how large or small the eigenvalues of $\mathbf{H}^{-1}$ are, we will amplify or attenuate our measurement's noise. The eigenvalues of $\mathbf{H}^{-1}$ are actually related to the eigenvalues of $\mathbf{H}$! Show that if λ is an eigenvalue of a matrix $\mathbf{H}$, then $\frac{1}{\lambda}$ is an eigenvalue of the matrix $\mathbf{H}^{-1}$.

Hint: Start with an eigenvalue λ and one corresponding eigenvector $\vec{v}$, such that they satisfy $\mathbf{H}\vec{v} = \lambda\vec{v}$.

Solution:

A is invertible,
$\Rightarrow A\vec{x} = \vec{b}$ has a unique solution for all $\vec{b}$.
$\Rightarrow A\vec{x} = \vec{0}$ has a unique solution.
$\Rightarrow \vec{x} = \vec{0}$ is the only solution to $A\vec{x} = \vec{0}$.
$\Rightarrow A\vec{x} = \vec{0}$ has no non zero vectors $\vec{x}$ that satisfy it.

Therefore, 0 is not an eigenvalue. Let $\vec{v}$ be the eigenvector of $\mathbf{A}$ corresponding to λ.

$$\mathbf{A}\vec{v} = \lambda\vec{v}$$

Since we know that $\mathbf{A}$ is invertible, we can left-multiply both sides by $\mathbf{A}^{-1}$.

$$\mathbf{A}^{-1}\mathbf{A}\vec{v} = \lambda\mathbf{A}^{-1}\vec{v}$$

$$\vec{v} = \lambda\mathbf{A}^{-1}\vec{v}$$

$$\mathbf{A}^{-1}\vec{v} = \frac{1}{\lambda}\vec{v}$$

(c) We are going to try different $\mathbf{H}$ matrices in this problem and compare how they deal with noise. Run all of the cells in the attached IPython notebook. Which matrix performs best in reconstructing the original image and why? What do you observe regarding the eigenvalues of matrices $\mathbf{H_1}, \mathbf{H_2}$ and $\mathbf{H_3}$? What special matrix is $\mathbf{H}_1$? Notice that each plot in the iPython notebook returns the result of trying to image a noisy image as well as the minimum absolute value of the eigenvalue of each matrix. Comment on the effect of small eigenvalues on the noise in the image.

Solution:

See `sol5.ipynb`.

Notice that we are printing the eigenvalue with the smallest absolute value. As the absolute value of the smallest eigenvalue of $\mathbf{H}$ decreases, the absolute value of the largest eigenvalue of $\mathbf{H^{-1}}$ increases (see why this happens in part d), hence the noise in the result increases. The matrix $\mathbf{H}_1$ is the identity matrix. Notice also that there are almost no visibile differences between the matrices $\mathbf{H}_2$ and $\mathbf{H}_3$.

(d) Now, because there is noise in our measurements, there will be noise in our recovered image. However, the noise is scaled. The noise in the recovered image, $\hat{\vec{w}}$, is related to $\vec{w}$, but it is transformed by $\mathbf{H}^{-1}$. Specifically,

$$\hat{\vec{w}} = \mathbf{H}^{-1}\vec{w} \tag{9}$$

To analyze how this transformation alters $\vec{w}$, consider representing $\vec{w}$ as a linear combination of the eigenvectors of $\mathbf{H}^{-1}$,

$$\vec{w} = \alpha_1\vec{b}_1 + \ldots + \alpha_N\vec{b}_N. \tag{10}$$

Where, $\vec{b}_i$ is $\mathbf{H}^{-1}$'s eigenvector corresponding to eigenvalue $\frac{1}{\lambda_i}$.

Show that we can express the recovered image's noise as,

$$\hat{\vec{w}} = \mathbf{H}^{-1}\vec{w} = \alpha_1 \frac{1}{\lambda_1}\vec{b}_1 + \ldots + \alpha_N \frac{1}{\lambda_N}\vec{b}_N \tag{11}$$

Depending on the size of the eigenvalues, noise in the recovered image will be amplified or attenuated. For eigenvectors with large eigenvalues, will the noise signal along those eigenvectors be amplified of attenuated? For eigenvectors with small eigenvalues, will the noise signal along those eigenvectors be amplified of attenuated?

Solution: To show this, we will write $\vec{w}$ as a linear combination of the eigenvectors of $\mathbf{H}^{-1}$ and then use the distributivity property of matrix-vector multiplication operation:

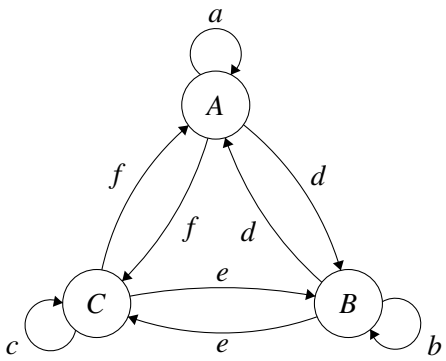
$$\begin{aligned}\hat{\vec{w}} = \mathbf{H}^{-1}\vec{w} &= \mathbf{H}^{-1}(\alpha_1\vec{b}_1 + \ldots + \alpha_N\vec{b}_N) \\ &= \alpha_1\mathbf{H}^{-1}\vec{b}_1 + \ldots + \alpha_N\mathbf{H}^{-1}\vec{b}_N \\ &= \alpha_1 \frac{1}{\lambda_1}\vec{b}_1 + \ldots + \alpha_N \frac{1}{\lambda_N}\vec{b}_N\end{aligned}$$

For eigenvectors with small eigenvalues, the noise signal will be amplified. This is bad and could corrupt the recovered image significantly.

For eigenvectors with large eigenvalues, the noise signal will be attenuated. This is better and will not corrupt the recovered image as much.

4. Reservoirs That Give and Take

Consider a network of three water reservoirs A, B, and C. At the end of each day, water transfers among the reservoirs according to the directed graph shown below.



The parameters a, b, and c—which label the self-loops—denote the *fractions* of the water in reservoirs A, B, and C, respectively, that stay in the same reservoir at the end of each day n. The parameters d, e, and f denote the fractions of the reservoir contents that transfer to adjacent reservoirs at the end of each day, according to the directed graph above.

Assume that the reservoir system is conservative—no water enters or leaves the system, which means that the total water in the network is constant. Accordingly, *for each node*, the weights on its self-loop and its two outgoing edges sum to 1; for example, for node A, we have

$$a + d + f = 1,$$

and similar equations hold for the other nodes. Moreover, assume that all the edge weights are positive numbers—that is,

$$0 < a, b, c, d, e, f < 1.$$

The state evolution equation governing the water flow dynamics in the reservoir system is given by

$$\vec{s}[n+1] = \mathbf{A}\vec{s}[n], \tag{12}$$

where the 3×3 matrix $\mathbf{A}$ is the state transition matrix, and

$$\vec{s}[n] = \begin{bmatrix} s_A[n] & s_B[n] & s_C[n] \end{bmatrix}^T \in \mathbb{R}^3 \tag{13}$$

is the nonnegative state vector that shows the water distribution among the three reservoirs at the end of day n, as fractions of the total water in the network.

(a) Determine the state transition matrix $\mathbf{A}$.

Solution:

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

(b) For some systems, there exists an equilibrium state —that is, a state for which the following is true:

$$\vec{s}[n+1] = \vec{s}[n] = \vec{s}^*$$

Determine if for the systems described above, there exists a equilibrium state. If such a state exists, find it.

Hint: What's special about the rows of this matrix?

Solution:

Notice for the above matrix, $\mathbf{A}^T = \mathbf{A}$. Such a matrix is called symmetric. This implies both the rows and the columns of this matrix sum to one.

Consider what happens when you apply this matrix to a uniform vector, $\vec{s}[0] = [x\ x\ x]^T = x[1\ 1\ 1]^T$.

$$\vec{s}[1] = \mathbf{A}\vec{s}[0] = \begin{bmatrix} a & d & f \\ d & b & e \\ f & e & c \end{bmatrix} \begin{bmatrix} x \\ x \\ x \end{bmatrix} = \begin{bmatrix} ax+dx+fx \\ dx+bx+ex \\ fx+ex+cx \end{bmatrix} = \begin{bmatrix} x(a+d+f) \\ x(d+b+e) \\ x(f+e+c) \end{bmatrix} = \begin{bmatrix} x \\ x \\ x \end{bmatrix}$$

Since $\vec{s}[1] = \vec{s}[0]$, $\vec{s}[0]$ is the equilibrium state $\vec{s}^*$.

An alternative way of doing the problem would be to compute the nullspace of $\mathbf{A} - \mathbf{I}$ in the following way.

$$\begin{bmatrix} a & d & f \\ d & b & e \\ f & e & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} a & d & f \\ d & b & e \\ f & e & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \mathbf{0}$$

$$\Rightarrow (\mathbf{A} - \mathbf{I}) [x_1 x_2 x_3]^T = \mathbf{0}$$

$$\begin{bmatrix} a-1 & d & f \\ d & b-1 & e \\ f & e & c-1 \end{bmatrix} \sim \begin{bmatrix} a-1 & d & f \\ d & b-1 & e \\ a+d+f-1 & b+d+e-1 & e+f+c-1 \end{bmatrix} \sim \begin{bmatrix} a-1 & d & f \\ d & b-1 & e \\ 0 & 0 & 0 \end{bmatrix}$$

Thus we can set x_3 as a free variable (say $x_3 = 1$), which gives us the following 2 equations for x_1, x_2

$$(a-1)x_1 + dx_2 + f = 0 = (-d-f)x_1 + dx_2 + f$$

$$dx_2 + (b-1)x_2 + e = 0 = dx_1 + (-d-e)x_2 + e$$

Adding the above two equations gives

$$f(1-x_1) + e(1-x_2) = 0$$

Thus

$$x_1 = x_2 = x_3 = 1$$

is a basis for the nullspace, and $[x\ x\ x]^T$ is a general solution.

(c) Suppose the state transition matrix for the network is given by

$$\mathbf{A} = \begin{bmatrix} \frac{1}{5} & \frac{2}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{1}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{2}{5} & \frac{1}{5} \end{bmatrix}.$$

Is it possible to determine the state $\vec{s}[n]$ from the subsequent state $\vec{s}[n+1]$? You do not have to find $\vec{s}[n]$ from $\vec{s}[n+1]$, just determine whether it is possible to do so.

Solution:

We row-reduce $\mathbf{A}$ and see what the pivots look like. If they're all nonzero, then $\mathbf{A}$ is invertible and reverse-time inference (obtaining $\vec{s}[n]$ from $\vec{s}[n+1]$) is possible. If any pivot is zero, then $\mathbf{A}$ is not invertible, in which case we cannot determine $\vec{s}[n]$ from $\vec{s}[n+1]$.

$$\begin{bmatrix} \frac{1}{5} & \frac{2}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{1}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{2}{5} & \frac{1}{5} \end{bmatrix} \sim \begin{bmatrix} \frac{1}{5} & \frac{2}{5} & \frac{2}{5} \\ 0 & -\frac{3}{5} & -\frac{2}{5} \\ 0 & -\frac{2}{5} & -\frac{3}{5} \end{bmatrix} \sim \begin{bmatrix} \frac{1}{5} & \frac{2}{5} & \frac{2}{5} \\ 0 & -\frac{3}{5} & -\frac{2}{5} \\ 0 & 0 & -\frac{1}{3} \end{bmatrix}$$

All three pivots are nonzero. Therefore $\mathbf{A}$ is invertible and $\vec{s}[n] = \mathbf{A}^{-1}\vec{s}[n+1]$. The problem does not ask us to compute $\mathbf{A}^{-1}$ but merely whether it's possible to determine $\vec{s}[n]$ from $\vec{s}[n+1]$. The answer is yes!

Note that in this problem we had the columns with entries that sum to 1 and all entries are between 0 and 1. It can be shown that under the first condition we are guaranteed the existence of a steady state. See practice problem 7 for a guided proof.

5. Power Iteration

Many linear algebra applications only require knowledge of an approximate value of the largest eigenvalue of a matrix, known as the dominant eigenvalue λ_D, as well as its corresponding eigenvector $\vec{v}_D$. However, in the case of large matrices, using the determinant formulation to find the eigenvalues is often computationally intractable. Thus, an algorithm known as **power iteration** is often used, which works as long as there is a single, unique dominant eigenvalue.

In all parts below, let $\mathbf{A}$ be a $n \times n$ matrix with n non-zero eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$, which are distinct in magnitude, corresponding to eigenvectors $\vec{v}_1, \vec{v}_2, ..., \vec{v}_n$ where

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > ... > |\lambda_n| \geq 0$$

(i.e. there is a single dominant eigenvalue, $\lambda_D = \lambda_1$).

(a) Let $\vec{b} \in \mathbb{R}^n$. Why are we guaranteed that there will exist coefficients $c_1, c_2, ..., c_n$ such that $\vec{b} = \sum_{i=1}^{n} c_i \vec{v}_i$?

Solution:
Since $\mathbf{A}$ has n distinct eigenvalues, we are guaranteed that that all n eigenvectors are linearly independent and thus form a basis in $\mathbb{R}^n$.

(b) Consider the quantity $\frac{1}{\lambda_1^k}\mathbf{A}^k\vec{b}$. Show that:

$$\lim_{k\to\infty} \frac{1}{\lambda_1^k}\mathbf{A}^k\vec{b} = c_1\vec{v}_1$$

Solution: We have that:

$$\begin{aligned}
\lim_{k\to\infty} \frac{1}{\lambda_1^k}\mathbf{A}^k\vec{b} &= \lim_{k\to\infty} \frac{1}{\lambda_1^k}\mathbf{A}^k(c_1\vec{v}_1 + \vec{v}_2\vec{v}_2 + ... + c_n\vec{v}_n) \\
&= \lim_{k\to\infty} \frac{1}{\lambda_1^k}(c_1\lambda_1^k\vec{v}_1 + c_2\lambda_2^k\vec{v}_2 + ... + c_n\lambda_n^k\vec{v}_n) \\
&= \lim_{k\to\infty} \left(c_1\vec{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\vec{v}_2 + ... + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\vec{v}_n \right) \\
&= c_1\vec{v}_1
\end{aligned}$$

(c) In part (b) above, we see that if we have the eigenvalue, we can find an eigenvector. However, we have neither the eigenvalue nor the eigenvector when we begin. To make the above process useful, we must somehow acquire the dominant eigenvalue. It turns out that because $\mathbf{A}^k\vec{b}$ can be approximated by $\lambda_1^k c_1\vec{v}_1$ for higher powers, if we can measure the length of this vector, we have information about λ_1^k. In certain circumstances, computing the length of $\mathbf{A}^k\vec{b}$ is easier than computing the eigenvalue directly. The definition of the length of a vector, usually called its *norm*, and a few of its properties are given as follows:

$$\text{Length of } \vec{v}\text{: } \|\vec{v}\| = \left\| \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \right\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

i. If $\|\vec{v}\| = 0$ then $\vec{v} = \vec{0}$.
ii. $\|\alpha\vec{v}\| = |\alpha|\|\vec{v}\|$.
iii. $\|\vec{v} + \vec{w}\| \leq \|\vec{v}\| + \|\vec{w}\|$.

Now show that:

$$\lim_{k\to\infty} \frac{\mathbf{A}^k\vec{b}}{||\mathbf{A}^k\vec{b}||} = \frac{c_1\vec{v}_1}{||c_1\vec{v}_1||}$$

This result shows that the given quantity will converge to the eigenvector of unit length corresponding to the largest eigenvalue (provided that $c_1 \neq 0$)

Solution: If we assume $\lambda_1 > 0$, we can multiply the numerator and denominator by $\frac{1}{\lambda_1^k}$, use the second property of the length, and then apply our result from part (b).

$$\lim_{k\to\infty} \frac{\mathbf{A}^k\vec{b}}{||\mathbf{A}^k\vec{b}||} = \lim_{k\to\infty} \frac{\frac{1}{\lambda_1^k}\mathbf{A}^k\vec{b}}{||\frac{1}{\lambda_1^k}\mathbf{A}^k\vec{b}||} = \frac{c_1\vec{v}_1}{||c_1\vec{v}_1||}$$

Therefore, for a sufficiently large k, we can approximate the dominant eigenvector as $\vec{v}_D \approx \frac{\mathbf{A}^k\vec{b}}{||\mathbf{A}^k\vec{b}||}$

If the eigenvalue is not positive, we will have an extra negative sign, however, this is not an issue as a scalar multiple of an eigenvector will still be an eigenvector.

(d) From the previous subpart, we can converge to the eigenvector corresponding to the largest eigenvalue provided that k is large enough. However, we are also interested in computing an approximation for the dominant eigenvalue. We do this via the following quantity:

$$R(\mathbf{A},\vec{v}) = \frac{\vec{v}^T\mathbf{A}\vec{v}}{\vec{v}^T\vec{v}}$$

In the context of this algorithm, we can approximate the dominant eigenvalue by substituting the approximate dominant eigenvector $\vec{v}_D$ into the expression. That is $\lambda_1 \approx R(\mathbf{A},\vec{v}_D)$. To see why this expression can approximate the dominant eigenvalue, determine the value of $R(\mathbf{A},\alpha\vec{v}_1)$, where $\alpha\vec{v}_1$ is an eigenvector corresponding to λ_1.

Solution: We have:

$$R(\mathbf{A},\alpha\vec{v}_1) = \frac{(\alpha\vec{v})_1^T\mathbf{A}(\alpha\vec{v}_1)}{(\alpha\vec{v}_1)^T(\alpha\vec{v}_1)} = \frac{\alpha^2\vec{v}_1^T\lambda_1\vec{v}_1}{\alpha^2\vec{v}_1^T\vec{v}_1} = \lambda_1$$

6. (Practice) Page Rank

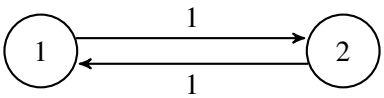
Learning Goal: *This problem highlights the use of transition matrices in modeling dynamical linear systems. Predictions about the steady state of a system can be made using the eigenvalues and eigenvectors of this matrix.*

In homework and discussion, we have discussed the behavior of water flowing in reservoirs and the people flowing in social networks. We now consider the setting of web traffic where the dynamical system can be described with a directed graph, also known as state transition diagram.

As we have seen in lecture and discussion the "transition matrix", $\mathbf{T}$, can be constructed using the state transition diagram, as follows: entries t_{ji}, represent the *proportion* of the people who are at website i that click the link for website j.

The steady-state frequency for a graph of websites is related to the eigenspace associated with eigenvalue 1 for the "transition matrix" of the graph. Once computed, an eigenvector with eigenvalue 1 will have values which correspond to the steady-state frequency for the fraction of people for each webpage. When this eigenvector's values are made to sum to one (to conserve people), the i^{th} element of the eigenvector corresponds to the fraction of people on the i^{th} website.

(a) For graph A shown below, what are the steady-state frequencies for the two webpages? Graph A has weights in place to help you construct the transition matrix.



Graph A - weighted

Solution:

To determine the steady-state frequencies for 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.

The transition matrix of graph A:

$$\mathbf{T} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{14}$$

To determine the eigenvalues of this matrix:

$$\det\left(\begin{bmatrix} -\lambda & 1 \\ 1 & -\lambda \end{bmatrix}\right) = \lambda^2 - 1 = 0 \tag{15}$$

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

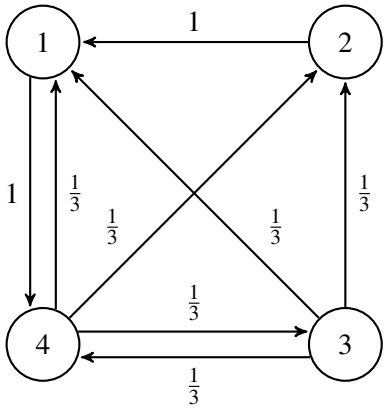
$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{16}$$

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 steady-state frequency eigenvector is $\begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$ and each webpage has a steady-state frequency of 0.5.

(b) For graph B, what are the steady-state frequencies for the webpages? You may use IPython and the Numpy command numpy.linalg.eig for this. Graph B is shown below, with weights in place to help you construct the transition matrix.



Graph B - weighted

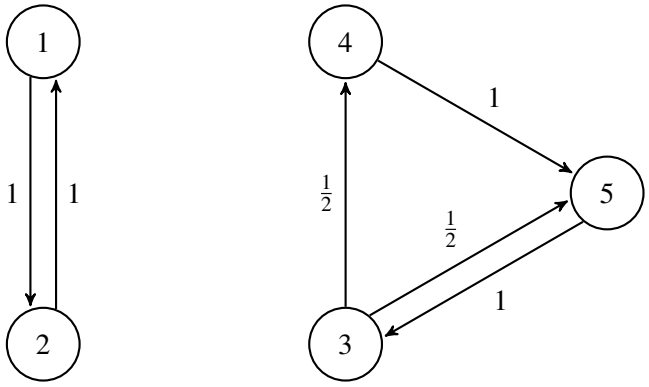
Solution:

To determine the steady-state frequencies, we need to create the transition matrix **T** first.

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

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 steady-state frequencies for the pages.

(c) Find the eigenspace that corresponds to the steady-state for graph C. How many independent systems (disjoint sets of webpages) are there in graph C versus in graph B? What is the dimension of the eigenspace corresponding to the steady-state for graph C? Again, graph C with weights in place is shown below. You may use IPython to compute the eigenvalues and eigenvectors again.



Graph C - weighted

Solution:

The transition matrix for graph C is

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

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 eigenspace associated with $\lambda = 1$ is span by the vectors $\begin{bmatrix}0 & 0 & 0.4 & 0.2 & 0.4\end{bmatrix}^T$ and $\begin{bmatrix}0.5 & 0.5 & 0 & 0 & 0\end{bmatrix}^T$. While, any linear combination of these vectors is a eigenvector, these two particular vectors have a nice interpretation.

The first eigenvector describes the steady-state frequencies for the last three webpages, and the second vector describes the steady-state frequencies for the first two webpages. This makes sense since there are essentially "two internets", or two disjoint sets of webpages. Surfers cannot transition between the two, so you cannot assign steady-state frequencies to webpage 1 and webpage 2 relative to the rest. This is why the eigenspace corresponding to the steady-state has dimension 2.

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

7. (Practice) Is There A Steady State?

So far, we've seen that for a conservative state transition matrix $\mathbf{A}$, we can find the eigenvector, $\vec{v}$, corresponding to the eigenvalue $\lambda = 1$. This vector is the steady state since $\mathbf{A}\vec{v} = \vec{v}$. However, we've so far taken for granted that the state transition matrix even has the eigenvalue $\lambda = 1$. Let's try to prove this fact.

(a) Show that if λ is an eigenvalue of a matrix $\mathbf{A}$, then it is also an eigenvalue of the matrix $\mathbf{A}^T$.

Hint: The determinants of $\mathbf{A}$ and $\mathbf{A}^T$ are the same. This is because the volumes which these matrices represent are the same.

Solution:

Recall that we find the eigenvalues of a matrix $\mathbf{A}$ by setting the determinant of $\mathbf{A} - \lambda\mathbf{I}$ to 0.

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \det\left((\mathbf{A} - \lambda\mathbf{I})^T\right) = \det\left(\mathbf{A}^T - \lambda\mathbf{I}\right) = 0$$

Since the two determinants are equal, the characteristic polynomicals of the two matrices must also be equal. Therefore, they must have the same eigenvalues.

(b) Let a square matrix $\mathbf{A}$ have, for each row, entries that sum to one. Show that $\vec{1} = \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^T$ is an eigenvector of $\mathbf{A}$. What is the corresponding eigenvalue?

Solution:

Recall that if the rows of $\mathbf{A}$ sum to one, then $\mathbf{A}\vec{1} = \vec{1}$. Therefore, the corresponding eigenvalue is $\lambda = 1$.

(c) Let's put it together now. From the previous two parts, show that any conservative state transition matrix will have the eigenvalue $\lambda = 1$. Recall that conservative state transition matrices have, for each column, entries that sum to 1.

Solution:

If we tranpose a conservative state transition matrix $\mathbf{A}$, then the rows of $\mathbf{A}^T$ (or the columns of $\mathbf{A}$) sum to one by definition of a conservative system. Then, from part (b), we know that $\mathbf{A}^T$ has the eigenvalue $\lambda = 1$. Furthermore, from part (a), we know that the $\mathbf{A}$ and $\mathbf{A}^T$ have the same eigenvalues, so $\mathbf{A}$ also has the eigenvalue $\lambda = 1$.

8. Homework Process and Study Group

Who else did you work with on this homework? List names and student ID's. (In case of homework party, you can also just describe the group.) How did you work on this homework?

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.

EECS16A: Homework 5

Problem 3: Noisy Images

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

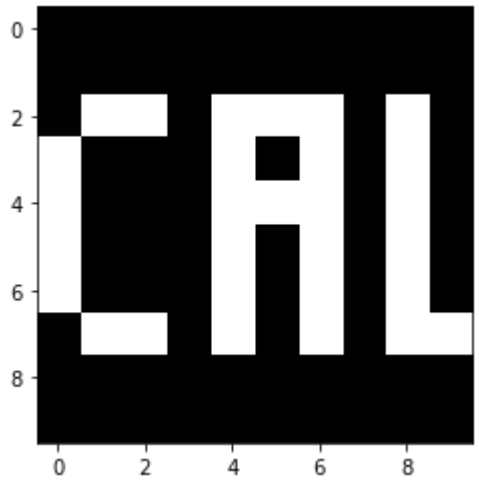
Let's load some data to start off with.

In [2]:
```
H3 = np.loadtxt("cond_10e6.txt", delimiter=',').reshape(100,100)
H2 = np.loadtxt("cond_1e3.txt", delimiter=',').reshape(100,100)
H1 = np.eye(100)
img = np.loadtxt("image.txt", delimiter=',').reshape(10,10)
```

The code below displays the image.

In [3]:
```
plt.figure(0)
plt.imshow(img,cmap='gray')
```

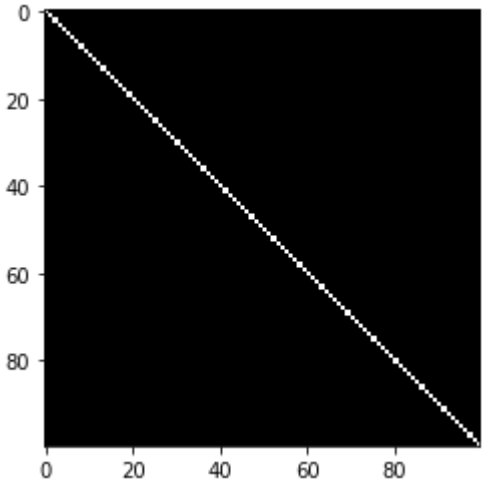
Out[3]:
```
<matplotlib.image.AxesImage at 0x281d9cbb240>
```

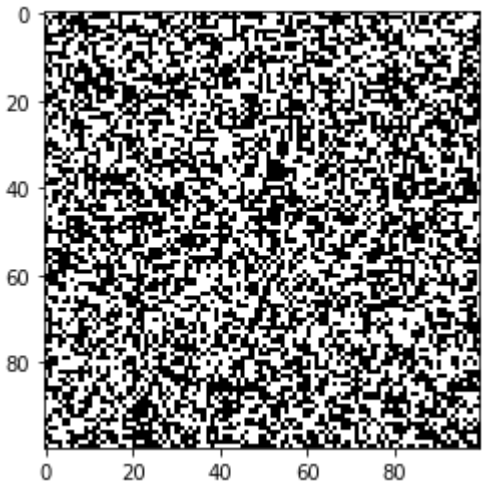


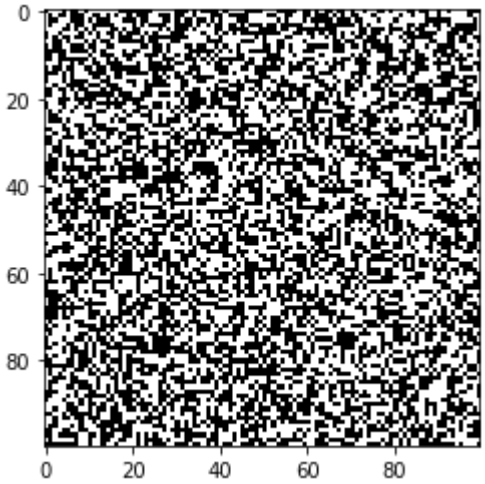
Then, lets display the set of masks

In [4]:
```
plt.figure(1)
plt.imshow(H1,cmap='gray')
plt.figure(2)
plt.imshow(H2,cmap='gray')
plt.figure(3)
plt.imshow(H3,cmap='gray')
```

Out[4]: <matplotlib.image.AxesImage at 0x281da026e48>







We'll use numpy.random to make some noise.

In [3]:
```
noise = np.random.normal(0.5,0.1)
```

Lets compute the $\vec{b}$ vector for each matrix and add some noise to the $\vec{b}$ vector.

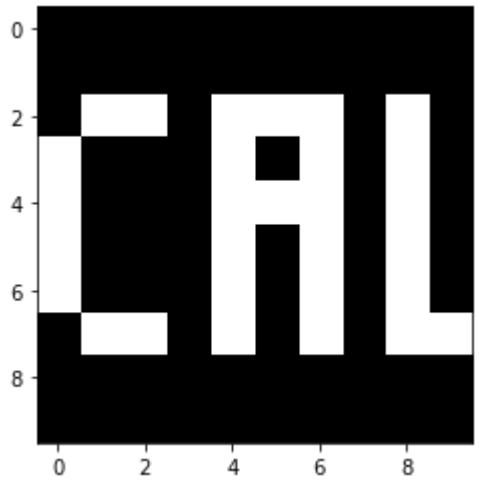
In [4]:
```
b1 = H1.dot(img.reshape(100)) + noise
b2 = H2.dot(img.reshape(100)) + noise
b3 = H3.dot(img.reshape(100)) + noise
```

First, let's compute $\vec{x}_1$ after adding noise and find the minimum eigenvalue of $\mathbf{H_1}$.

In [5]:
```
x1 = np.linalg.inv(H1).dot(b1)
eigenvalues1 = np.linalg.eig(H1)[0]
print("Is the matrix invertible?", abs(np.linalg.det(H1)) > 0.5)
print("The smallest eigenvalue is:", min(np.absolute(eigenvalues1)))
print("Number of eigenvectors:", len(eigenvalues1))
plt.imshow(x1.reshape(10,10), cmap='gray')
```

```
Is the matrix invertible? True
The smallest eigenvalue is: 1.0
Number of eigenvectors: 100
```

Out[5]: <matplotlib.image.AxesImage at 0x1da324aaeb8>

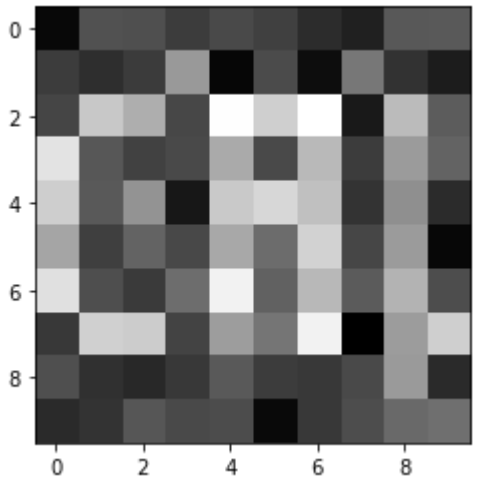


Now let's compute $\vec{x}_2$ and find the minimum eigenvalue of $\mathbf{H_2}$.

In [8]:
```
x2 = np.linalg.inv(H2).dot(b2)
eigenvalues2 = np.linalg.eig(H2)[0]
print("Is the matrix invertible?", abs(np.linalg.det(H2)) > 0.5)
print("The smallest eigenvalue is:", min(np.absolute(eigenvalues2)))
print("Number of eigenvectors:", len(eigenvalues2))
plt.imshow(x2.reshape(10,10), cmap='gray')
```

```
Is the matrix invertible? True
The smallest eigenvalue is: 0.29516363308630444
Number of eigenvectors: 100
```

Out[8]: <matplotlib.image.AxesImage at 0x281da9e9908>

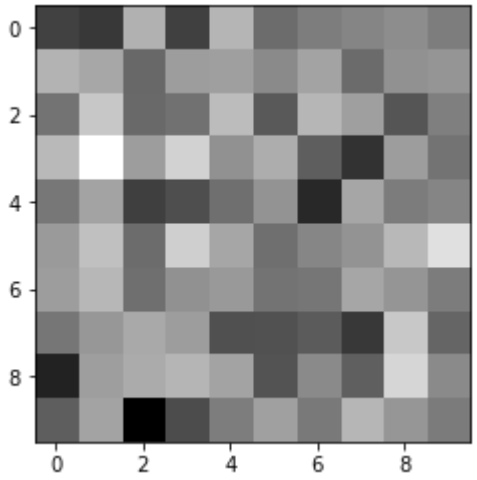


Now let's compute $\vec{x}_3$ and find the minimum eigenvalue of $\mathbf{H}_3$.

In [9]:
```
x3 = np.linalg.inv(H3).dot(b3)
eigenvalues3 = np.linalg.eig(H3)[0]
print("Is the matrix invertible?", abs(np.linalg.det(H3)) > 0.5)
print("The smallest eigenvalue is:", min(np.absolute(eigenvalues3)))
print("Number of eigenvectors:", len(eigenvalues3))
plt.imshow(x3.reshape(10,10), cmap='gray')
```

```
Is the matrix invertible? True
The smallest eigenvalue is: 1.2184217529708448e-05
Number of eigenvectors: 100
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x105510a10>
```



Problem 6: Page Rank

In [ ]:
```
# Though it is not required you may use iPython for your calculations in pa
```