Linear Data Chapter 11

Written by: Emily J. King

1. Assume that $\mathbf{C}$ is a 4×4 matrix, where

$$\vec{y} = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 0 \end{pmatrix} \in \mathcal{E}_{-2}(\mathbf{C})$$

Explicitly compute $\mathbf{C}^3\vec{y}$.

$$\mathbf{C}^3\vec{y} = \mathbf{C}^2(\mathbf{C}\vec{y}) = \mathbf{C}^2(-2\vec{y}) = -2\mathbf{C}^2\vec{y} = -2\mathbf{C}(\mathbf{C}\vec{y}) = (-2)^2\mathbf{C}\vec{y} = (-2)^3\vec{y} = -8\vec{y} = \begin{pmatrix} 0 \\ -8 \\ -16 \\ 0 \end{pmatrix}$$

2. Consider the following matrices

$$\mathbf{M} = \begin{pmatrix} 4 & 4 & 0 & 6 \\ 4 & -6 & 1 & 0 \\ 0 & 1 & -4 & 5 \\ 6 & 0 & 5 & -6 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

where $\mathbf{M}$ is invertible. Define $\mathbf{A} = \mathbf{M}\mathbf{D}\mathbf{M}^{-1}$. You must work all of the following problems by hand with justification to receive credit.

(a) Explicitly list the eigenvalues of $\mathbf{A}$.
The eigenvalues are just the diagonal elements of $\mathbf{D}$*. That is,* $-2, -1, 3, 0$.

(b) Explicitly give all elements of the null space of $\mathbf{A}$.
We have that

$$\text{null}(\mathbf{A}) = \left\{ \vec{x} \in \mathbb{R}^4 \mid \mathbf{A}\vec{x} = \vec{0} \right\} = \mathcal{E}_0(\mathbf{A}).$$

Thus, the column of $\mathbf{M}$ *corresponding to the* 0*th diagonal element of* $\mathbf{D}$ *is a basis for the null space of* $\mathbf{A}$*:*

$$\text{null}(\mathbf{A}) = \text{span}\left\{ \begin{pmatrix} 6 \\ 0 \\ 5 \\ -6 \end{pmatrix} \right\}.$$

(c) Explicitly compute the rank of $\mathbf{A}$ by hand.
By the rank-nullity theorem, the rank plus nullity is equal to the number of columns. From part (b), we have that the nullity (dimension of the nullspace) is 1. The number of columns of $\mathbf{A}$ *is 4. Thus, the rank of* $\mathbf{A}$ *is* $4 - 1 = 3$.

(d) Explicitly compute the determinant of $\mathbf{A}$ by hand.
[METHOD ONE] By part (b), $\mathbf{A}$ *isn't invertible because it isn't injective / one-to-one. Thus,* $\det(\mathbf{A}) = 0$.
[METHOD TWO] By part (c), $\mathbf{A}$ *isn't invertible because it isn't surjective / onto. Thus* $\det(\mathbf{A}) = 0$.
[METHOD THREE] We know that the determinant of the product of matrices is the product of the determinants. We also know that the determinant of a diagonal matrix is the product of the diagonal elements. Thus,

$$\det(\mathbf{A}) = \det(\mathbf{M}\mathbf{D}\mathbf{M}^{-1}) = \det(\mathbf{M})\det(\mathbf{D})\det(\mathbf{M}^{-1}) = \frac{\det(\mathbf{M})}{\det(\mathbf{M})}\det(\mathbf{D}) = (-2)(-1)(3)(0) = 0.$$

(e) Let $\mathbf{I}$ be the 4×4 identity matrix. Explicitly compute $(\mathbf{A}+2\mathbf{I})(\mathbf{A}+\mathbf{I})(\mathbf{A}-3\mathbf{I})\mathbf{A}$.

Since the eigenvalues of $\mathbf{A}$ *are* $-2, -1, 3, 0$*, the characteristic polynomial of* $\mathbf{A}$ *is* $(x+2)(x+1)(x-3)x$*, which means that* $(\mathbf{A}+2\mathbf{I})(\mathbf{A}+\mathbf{I})(\mathbf{A}-3\mathbf{I})\mathbf{A}$ *is the* 4×4 *zero matrix.*

(f) Let $\vec{y} \in \mathbb{R}^4$ be a random vector and let $\mathbf{U}$ be a random 4×4 orthogonal matrix. If you were to perform power iteration on the symmetric matrix $\mathbf{U}\mathbf{D}\mathbf{U}^\top$ (that is, multiply the vector by $\mathbf{U}\mathbf{D}\mathbf{U}^\top$, normalize, then take that vector multiply by $\mathbf{U}\mathbf{D}\mathbf{U}^\top$ and normalize, and repeat this process) for 1000 iterations, what is most likely to be the output? Fully justify your answer without any reference to Matlab/Python.
The largest (in absolute value) eigenvalue of $\mathbf{U}\mathbf{D}\mathbf{U}^\top$ *is 3, which occurs in the third diagonal position of* $\mathbf{D}$*. Thus, power iteration should converge a unit norm element of the span of the third column* $\vec{u}_3$ *of* $\mathbf{U}$.

$$\mathcal{E}_3(\mathbf{U}\mathbf{D}\mathbf{U}^\top) = \text{span}\left\{\vec{u}_3\right\}.$$

So, power iteration should return either

$$\frac{\vec{u}_3}{\|\vec{u}_3\|} \quad \textit{or} \quad \frac{-\vec{u}_3}{\|\vec{u}_3\|}.$$

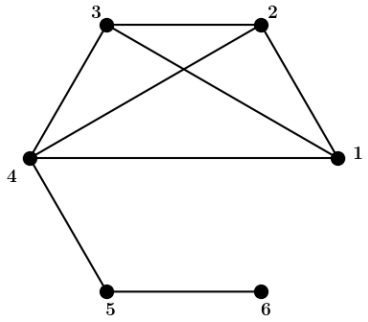
3. Are all square matrices diagonalizable? If so, explain why. If not, give an example of a non-diagonalizable matrix.

No, they aren't. Some possible examples of non-diagonalizable matrices we've seen include shear matrices like

$$\mathbf{S} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \text{ (from lecture) or } \begin{pmatrix} 1 & 0 \\ -3 & 1 \end{pmatrix} \text{ (from lab)}$$

where $\mathcal{E}_1(\mathbf{S})$ is the x-axis, a one-dimensional subspace of $\mathbb{R}^2$ (for the left-hand matrix) or $\mathcal{E}_1(\mathbf{S})$ is the y-axis, a one-dimensional subspace of $\mathbb{R}^2$ (for the right-hand matrix) and there are not other eigenspaces. Another example we've seen would be any rotation matrix where the rotation isn't either not doing anything or flipping the vector (e.g., the 2×2 matrix that rotates $\mathbb{R}^2$ by $30°$ counterclockwise about the origin).

4. Consider the graph/network below.



(a) Explicitly give the graph Laplacian $\mathbf{L}$ of the graph/network.

$$\mathbf{L} = \begin{pmatrix} 3 & -1 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 \\ -1 & -1 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

(b) Explicitly give (the entries of) a non-zero vector in the nullspace of this graph Laplacian $\mathbf{L}$. Fully justify your answer without any reference to Matlab/Python.

The all-ones vector

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

is in the nullspace of the graph Laplacian since constant vectors are always in the nullspace of a graph Laplacian of an undirected (weighted or unweighted) graph. Alternatively, one could note that the rows all sum to 0.
(More generally,

$$\text{null}(\mathbf{L}) = \text{span}\left\{\begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix}\right\} = \left\{\begin{pmatrix}c\\c\\c\\c\\c\\c\end{pmatrix} : c \in \mathbb{R}\right\}$$

)

(c) Which of the following is most likely to be a Fielder eigenvector of the graph above? Fully justify your answer without any reference to Matlab/Python.

A. $\begin{pmatrix}-0.3151\\-0.3151\\-0.3151\\-0.1620\\0.3759\\0.7312\end{pmatrix}$ B. $\begin{pmatrix}-0.3151\\0.3151\\0.3151\\0.1620\\0.3759\\0.7312\end{pmatrix}$ C. $\begin{pmatrix}-0.3151\\-0.3151\\-0.3151\\-0.1620\\-0.3759\\-0.7312\end{pmatrix}$ D. $\begin{pmatrix}-0.3151\\-0.3151\\0.3151\\0.1620\\-0.3759\\-0.7312\end{pmatrix}$

The signs of the entries of the Fiedler eigenvector give us a way to cluster the data. By inspection, vertices/nodes 1*,* 2*,* 3*, and* 4 *have lots of connections between each other (actually, the most they can possibly have); so, one could expect them to be in the same cluster. On the other hand, vertices/nodes* 5 *and* 6 *are not as well connected to the other vertices. In option [A.], vertices/nodes* 1-4 *are in one cluster and* 5 *and* 6 *are in another. This makes a lot of sense.*
Let's also look at the other options. In [B.], vertex/node 1 *is in one cluster by itself. That makes no sense. In [C.], all of the vertices/nodes are in the same cluster. Again, that doesn't make sense given what the graph/network looks like. Finally, in [D.] vertices/nodes* 1 *and* 2 *are clustered with* 5 *and* 6 *but not* 3 *and* 4*, which again makes no sense for this graph/network.*
So, [A.] is the only reasonable answer.

5. Consider the following (left) stochastic matrix

$$\mathbf{P} = \begin{pmatrix}1 & 1/3 & 1/5\\0 & 2/3 & 2/5\\0 & 0 & 2/5\end{pmatrix}$$

(a) Recall that if $\mathbf{Q}$ is a $d \times d$ (left) stochastic matrix and $\vec{e}_i$ is the ith standard basis vector in $\mathbb{R}^d$, then $\mathbf{Q}\vec{e}_i$ is a probability vector where the jth coordinate is the

probability of the system transitioning from state i to state j.
If a system explained by $\mathbf{P}$ is in state 3, what are the probabilities of it transitioning to state 1 or state 2 or remaining in state 3?
The probabilities of state 3's transition are

$$\mathbf{P}\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}1/5\\2/5\\2/5\end{pmatrix},$$

i.e., 1/5 probability of transitioning to state 1, 2/5 probability of transitioning to state 2, 2/5 probability of remaining in state 3.

(b) Give and run the Matlab/Python commands to compute the eigenvectors and eigenvalues of $\mathbf{P}$.
Matlab:

```
P=[1 1/3 1/5; 0 2/3 2/5; 0 0 2/5]
[U D] = eig(P)
```

Python:

```
import numpy as np
from numpy.linalg import eig
P=np.array([[1, 1/3, 1/5], [0, 2/3, 2/5], [0, 0, 2/5]])
d, U = eig(P)
d
U
```

(c) Given your answer to (b), give the steady state vector of $\mathbf{P}$.
A steady state vector of a (left) stochastic matrix is an eigenvector for eigenvalue 1 which is also a probability vector (i.e., all entries are non-negative and sum to 1). The computation above tells us that

$$\mathcal{E}_1(\mathbf{P}) = \text{span}\left\{\begin{pmatrix}1\\0\\0\end{pmatrix}\right\}.$$

The only vector in $\mathcal{E}_1(\mathbf{P})$ that has non-negative entries summing to 1 is just

$$\begin{pmatrix}1\\0\\0\end{pmatrix}$$

(d) Explain why intuitively your answer to (c) makes sense.
This makes sense because once the system is in state 1, it doesn't leave state 1. I.e.,

$$\mathbf{P}\begin{pmatrix}1\\0\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\end{pmatrix}.$$

And if the system is in states 2 *or* 3*, there is some probability of going to state* 1*. Thus, in the long run, everything eventually ends up in state* 1*.*