Soraya Terrab
August 30, 2020

Figure 1: Somewhere on Aspen Mountain this past June

Part 1: Short Bio

My name is Soraya and I am a second-year PhD student in Applied Math. I am from Morocco, but I have lived in the US since 2009. I enjoy cooking and hiking, but I would say I'm a less frequent hiker than the typical Coloradan. I used to be a manufacturing engineer and quality manager before I transitioned to applied math grad school. It's actually while I was a manufacturing engineer that I got first introduced to computer vision through Cognex machine systems, which we used to measure product specifications of tiny parts. That experience planted a seed that I didn't realize would grow over time. During my Master's at the University of Washington, I used ML to automate visual inspection of a simulated product for a course final project. Now at Mines, I would like to focus my PhD dissertation in the area of Imaging Science, and potentially consider problems that arise in medical imaging. I hope this course will provide the application tools that will guide my research in image processing.

Part 2: Piazza Post

See `https://piazza.com/class/kdjcm49icok4ei?cid=42`

Part 3: Math Review

1. Consider the matrix

$$A = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}$$

a. Compute the determinant of the matrix, $|A|$.
$\det A = |A| = (4)(1) - (-2)(1) = \boxed{6}$.

b. Compute the trace of the matrix.
$tr A = \text{sum of diagonal elements of } A = 4 + 1 = \boxed{5}$.

c. Which of the following matrices is the inverse of A?

(i) $A^{-1} = \begin{bmatrix} 1/4 & -1/2 \\ 1 & 1 \end{bmatrix}$

(ii) $A^{-1} = \begin{bmatrix} 4 & 1 \\ -2 & 1 \end{bmatrix}$

(iii) $A^{-1} = \begin{bmatrix} 1/6 & 1/3 \\ -1/6 & 2/3 \end{bmatrix}$

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} 1 & 2 \\ -1 & 4 \end{bmatrix} = \frac{1}{6}\begin{bmatrix} 1 & 2 \\ -1 & 4 \end{bmatrix} = \begin{bmatrix} 1/6 & 1/3 \\ -1/6 & 2/3 \end{bmatrix} \equiv \boxed{(iii)}$$

d. Which of the following vectors is the eigenvector of A?

(i) $\boldsymbol{x} = (-1\ 2)^T$

(ii) $\boldsymbol{x} = (2\ 1)^T$

(iii) $\boldsymbol{x} = (0\ 1)^T$

(iv) $\boldsymbol{x} = (1\ 0)^T$

We need to find $\boldsymbol{x}$ from the above such that $A\boldsymbol{x} = \lambda\boldsymbol{x}$, where $\lambda \in \mathbb{R}$ is the corresponding eigenvalue for that eigenvector $\boldsymbol{x}$. Instead of using the characteristic polynomial to compute the eigenvalues and their respective eigenvectors, we can compute $A\boldsymbol{x}$, for each of the four options above, and verify if it is a scalar multiple of that given $\boldsymbol{x}$.

(i) $A\boldsymbol{x} = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} -8 \\ 1 \end{bmatrix} \neq \lambda \begin{bmatrix} -1 \\ 2 \end{bmatrix}, \ \lambda \in \mathbb{R}$

(ii) $A\boldsymbol{x} = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 3 \end{bmatrix} = 3\begin{bmatrix} 2 \\ 1 \end{bmatrix} \implies \lambda = 3$

(iii) $A\boldsymbol{x} = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \neq \lambda \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \lambda \in \mathbb{R}$

(iv) $A\boldsymbol{x} = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \neq \lambda \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ \lambda \in \mathbb{R}$

Hence, $\boxed{(ii)\ \boldsymbol{x} = (2\ 1)^T}$ is the eigenvector of A.

Soraya Terrab
August 30, 2020

e. What is the corresponding eigenvalue?
As found above, $\boxed{3}$ is the corresponding eigenvalue.

2. Consider the matrix $B = \begin{bmatrix} 3 & 4 \\ 5 & -1 \end{bmatrix}$

   (a) Compute $(AB)^T$
   $$(AB)^T = \left(\begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 3 & 4 \\ 5 & -1 \end{bmatrix}\right)^T = \left(\begin{bmatrix} 2 & 18 \\ 8 & 3 \end{bmatrix}\right)^T = \boxed{\begin{bmatrix} 2 & 8 \\ 18 & 3 \end{bmatrix}}$$

   (b) Compute $B^T A^T$
   By definition, $(AB)^T = B^T A^T$, so $B^T A^T = \boxed{\begin{bmatrix} 2 & 8 \\ 18 & 3 \end{bmatrix}}$.

3. Consider the vectors $\boldsymbol{x} = (1\ 2\ 3)^T$ and $\boldsymbol{y} = (-1\ 2 - 3)^T$.

   a. Compute the inner (dot) product, $\boldsymbol{x} \cdot \boldsymbol{y}$.
   $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^T \boldsymbol{y} = (1\ 2\ 3) \cdot (-1\ 2 - 3)^T = (1)(-1) + (2)(2) + (3)(-3) = \boxed{-6}$

   b. Compute the vector (cross) product, $\boldsymbol{x} \times \boldsymbol{y}$.
   $$\boldsymbol{x} \times \boldsymbol{y} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ 1 & 2 & 3 \\ -1 & 2 & -3 \end{vmatrix} = \hat{i}(-6-6) - \hat{j}(-3+3) + \hat{k}(2+2) = \boxed{(-12\ 0\ 4)^T}$$

4. The faces of a 10-sided die are numbered 0 through 9.

   (a) If the die is rolled, what is the probability that the value of the roll is a prime number?
   The prime numbers in $\{0, \ldots, 9\}$ are 2, 3, 5, and 7. Given 4 possible prime numbers, the probability that a roll results in a prime number is $\frac{4}{10} = \boxed{0.4}$.

   (b) What is the expected value of the roll?
   The expected value, or the average value across all outcomes, is 4.5 given there is equal likelihood that 4 and 5 (the middle values) are achieved. To show it computationally, the average value can be found as: $\sum_{i=0}^{9} iP(i) = \frac{1}{10}\sum_{i=0}^{9} i = \frac{45}{10} = \boxed{4.5}$.

   (c) If the die is rolled twice, what is the probability that the same number is obtained both times?
   The probability for any of the numbers to be rolled is $\frac{1}{10}$. The probability for the same number to be obtained both times is the product of the probabilities for each event, which is $\frac{1}{10} \times \frac{1}{10} = \boxed{1\%}$.

5. Pull a card at random from a deck of cards. What is the conditional probability that the card is the ace of clubs, given that it is a black card?
I must admit that I am not too familiar with cards. So looking at the breakdown

from the Standard 52-card deck Wikipedia page (https://en.wikipedia.org/wiki/Standard_52-card_deck), if the card is black, then it is either a club or a spade (13 cards each, for a total of 26 possible black cards). There is only 1 ace of clubs, so the conditional probability to land an ace of clubs if the card is black is $\boxed{\frac{1}{26}}$.

6. A company makes widgets from three machines. Machine M1 makes 3000/hour, and 80% are good. Machine M2 makes 4000/hour, and 90% are good. Machine M3 makes 3000/hour, and 60% are good. All widgets are mixed together. What is the probability that a widget drawn at random is good (hint: use marginalization)?
   I'm not sure what is meant by marginalization here, but this is how I would approach the problem:
   For any given number n of production hours, the probability of good widgets is given as the total number of good widgets per all widgets produced at that given time, or

$$\frac{0.8(3000)n + 0.9(4000)n + 0.6(3000)n}{(3000)n + (4000)n + (3000)n} = \frac{2400 + 3600 + 1800}{10,000} = \frac{7800}{10,000} = \boxed{0.78}.$$

7. A medical test shows that a person has a disease. What is the probability that the person actually has the disease (hint: use Bayes' rule)? Here's what we know about the disease and the test:

   - 1 in 100 people have the disease. That is, if D is the event that a randomly selected individual has the disease, then $P(D) = 0.01$.
   - If H is the event that a randomly selected individual is healthy, then $P(H) = 0.99$.
   - If a person has the disease, then the probability that the blood test comes back positive is 0.95. That is, $P(T^+|D) = 0.95$.
   - If a person is healthy, then the probability that the diagnostic test comes back negative is 0.95. That is, $P(T^-|H) = 0.95$.
     Bayes' rule (Ref: https://en.wikipedia.org/wiki/Bayes%27_theorem) is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

   With the information we have, we want to know $P(D|T^+)$, or the probability that the person with the positive diagnostic test (T^+) actually has the disease D. Using Bayes' rule where we assume all negative tests confirm good health, i.e.

$P(H|T^-) = 1$,

$$\begin{aligned} P(D|T^+) &= \frac{P(T^+|D)P(D)}{P(T^+)}, \\ &= \frac{P(T^+|D)P(D)}{1 - P(T^-)}, \\ &= \frac{P(T^+|D)P(D)}{1 - \frac{P(T^-|H)P(H)}{P(H|T^-)}}, \\ &= \frac{(0.95)(0.01)}{1 - \frac{(0.95)(0.99)}{1}}, \\ &= \frac{0.0095}{1 - 0.9405}, \\ &= \frac{0.0095}{0.0595}, \\ &\approx \boxed{0.1597}. \end{aligned}$$

I worked out all problems as I was typing up the problem statements and the solutions. All of that took me about 120 minutes, with the last question taking up a good portion of the time. I might need more background on Bayes' rule, but it seems to me that we need more information on the reliability of the testing method, which I tried to overcome with the assumption I made on $P(H|T^-)$ to get $P(T^-)$.

Part 4: Python Programming

1. Use Numpy to create matrices A and B, with the values given in the "Math Review" section above. Calculate and print out all the quantities that you calculated by hand, in questions 1 through 3 above.

```python
import numpy as np
import sys

sys.stdout = open("HW0_Part1_output.txt", "w")

### Question 1:
A = np.array([[4, -2], [1, 1]])
print("===QUESTION 1 ANSWERS===")

det_A = np.linalg.det(A)
print("The determinant of A is %1.1f." % det_A)

trace_A = np.trace(A)
```

```python
print("The trace of A is %1.1f." % trace_A)

inv_A = np.linalg.inv(A)
print("The inverse of A is: ")
print(inv_A)

evals, evectors = np.linalg.eig(A)
evector1_nonnormalized = evectors[:,0]/min(evectors[:,0])
print("One of the eigenvectors of A  is: ")
print(evector1_nonnormalized[:,None])
print("Its corresponding eigenvalue is %1.1f." % evals[0])

### Question 2:

B = np.array([[3, 4], [5, -1]])
print("===QUESTION 2 ANSWERS===")

print("(AB)^T : ")
print(np.transpose(A@B))

print("B^TA^T : ")
print(np.transpose(B)@np.transpose(A))

### Question 3:

x = np.array([1,2,3])
y = np.array([-1,2,-3])
print("===QUESTION 3 ANSWERS===")

xdoty = np.inner(x,y)
print("The inner product of x and y is %1.1f." %xdoty)

print("The vector product of x and y is: ")
print(np.cross(x,y))

sys.stdout.close()
```

Listing 1: Output File

```
===QUESTION 1 ANSWERS===
The determinant of A is 6.0.
The trace of A is 5.0.
The inverse of A is:
[[ 0.16666667  0.33333333]
 [-0.16666667  0.66666667]]
One of the eigenvectors of A  is:
[[2.]
 [1.]]
Its corresponding eigenvalue is 3.0.
```

```
===QUESTION 2 ANSWERS===
(AB)^T :
[[ 2  8]
 [18  3]]
B^TA^T :
[[ 2  8]
 [18  3]]
===QUESTION 3 ANSWERS===
The inner product of x and y is -6.0.
The vector product of x and y is:
[-12   0   4]
```

2. Write a function that takes the parameters of a normal distribution (mean and sigma), and fills a 9×9 matrix with numbers from that distribution, using two "for" loops. Verify that the sum of all values is (very close) to 1.0.

```
import numpy as np
import sys


def gaussian9by9matrix(xymean, sigma):
    g = np.zeros((9, 9))
    x = np.linspace(xymean[0] - 4, xymean[0] + 4, 9)
    y = np.linspace(xymean[1] - 4, xymean[1] + 4, 9)
    for i in range(9):
        for j in range(9):
            # using 2D Gaussian distribution as given in ...
                https://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm
            g[i, j] = np.exp(-((x[i]-xymean[0])** 2 + ...
                (y[j]-xymean[1])** 2) / (2 * sigma ** 2)) / (2 * ...
                np.pi * sigma** 2)

    # Verify sum of matrix elements add up to 1
    if round(np.sum(g)) == 1:
        print("Great! For mean = (%d,%d) and sigma = %f, the sum ...
            of the matrix values is %f, close to 1." %(xymean[0], ...
            xymean[1], sigma, np.sum(g)))
    elif round(np.sum(g)) > 1:
        print("For mean = (%d,%d) and sigma = %f, the sum of the ...
            matrix values is %f. Try again with a larger sigma." ...
            %(xymean[0], xymean[1], sigma, np.sum(g)) )
    else:
        print("For mean = (%d,%d) and sigma = %f, the sum of the ...
            matrix values is %f. Try again with a smaller sigma." ...
            %(xymean[0], xymean[1], sigma, np.sum(g)))
    return g


sys.stdout = open("HW0_Part2_output.txt", "w")
```

```python

print("==== Sigma = 1 ====")
G = gaussian9by9matrix((0, 0), 1)
print("Output matrix for mean=(0,0) and sigma = 1:")
print(G)

print("Mean values different from (0,0), verifying that matrix ...
    elements still add up to 1:")
G = gaussian9by9matrix((2, 2), 1)
G = gaussian9by9matrix((-1, 2), 1)

print("==== Sigma != 1 ====")
gaussian9by9matrix((0, 0), 5)
gaussian9by9matrix((0, 0), 0.1)

print("Mean values different from (0,0), verifying that matrix ...
    elements do not add up to 1 when sigma != 1:")
G = gaussian9by9matrix((2, 2), 5)
G = gaussian9by9matrix((-1, 2), 0.1)

sys.stdout.close()
```

Listing 2: Output File

```
==== Sigma = 1 ====
Great! For mean = (0,0) and sigma = 1.000000, the sum of the ...
    matrix values is 0.999994, close to 1.
Output matrix for mean=(0,0) and sigma = 1:
[[1.79105293e-08 5.93115274e-07 7.22562324e-06 3.23829967e-05
  5.33905355e-05 3.23829967e-05 7.22562324e-06 5.93115274e-07
  1.79105293e-08]
 [5.93115274e-07 1.96412803e-05 2.39279779e-04 1.07237757e-03
  1.76805171e-03 1.07237757e-03 2.39279779e-04 1.96412803e-05
  5.93115274e-07]
 [7.22562324e-06 2.39279779e-04 2.91502447e-03 1.30642333e-02
  2.15392793e-02 1.30642333e-02 2.91502447e-03 2.39279779e-04
  7.22562324e-06]
 [3.23829967e-05 1.07237757e-03 1.30642333e-02 5.85498315e-02
  9.65323526e-02 5.85498315e-02 1.30642333e-02 1.07237757e-03
  3.23829967e-05]
 [5.33905355e-05 1.76805171e-03 2.15392793e-02 9.65323526e-02
  1.59154943e-01 9.65323526e-02 2.15392793e-02 1.76805171e-03
  5.33905355e-05]
 [3.23829967e-05 1.07237757e-03 1.30642333e-02 5.85498315e-02
  9.65323526e-02 5.85498315e-02 1.30642333e-02 1.07237757e-03
  3.23829967e-05]
 [7.22562324e-06 2.39279779e-04 2.91502447e-03 1.30642333e-02
  2.15392793e-02 1.30642333e-02 2.91502447e-03 2.39279779e-04
  7.22562324e-06]
```

```
 [5.93115274e-07 1.96412803e-05 2.39279779e-04 1.07237757e-03
  1.76805171e-03 1.07237757e-03 2.39279779e-04 1.96412803e-05
  5.93115274e-07]
 [1.79105293e-08 5.93115274e-07 7.22562324e-06 3.23829967e-05
  5.33905355e-05 3.23829967e-05 7.22562324e-06 5.93115274e-07
  1.79105293e-08]]
Mean values different from (0,0), verifying that matrix elements ...
    still add up to 1:
Great! For mean = (2,2) and sigma = 1.000000, the sum of the ...
    matrix values is 0.999994, close to 1.
Great! For mean = (-1,2) and sigma = 1.000000, the sum of the ...
    matrix values is 0.999994, close to 1.
==== Sigma != 1 ====
For mean = (0,0) and sigma = 5.000000, the sum of the matrix ...
    values is 0.400284. Try again with a smaller sigma.
For mean = (0,0) and sigma = 0.100000, the sum of the matrix ...
    values is 15.915494. Try again with a larger sigma.
Mean values different from (0,0), verifying that matrix elements ...
    do not add up to 1 when sigma != 1:
For mean = (2,2) and sigma = 5.000000, the sum of the matrix ...
    values is 0.400284. Try again with a smaller sigma.
For mean = (-1,2) and sigma = 0.100000, the sum of the matrix ...
    values is 15.915494. Try again with a larger sigma.
```