Introduction to Mathematics for Data Science
Personal Assignnment 2

Zehao Qian

November 29, 2023

---

1 Question 1

1.1 Solve Matrix B

Consider the matrix $A = \begin{pmatrix} 5 & 2 & 3 \\ -2 & 3 & 1 \\ 0 & 2 & 2 \\ 1 & 0 & 1 \end{pmatrix}$ Find a matrix B such that $AB = \begin{pmatrix} 8 & 6 \\ -1 & 9 \\ 2 & 6 \\ 2 & 0 \end{pmatrix}$.

1.2 Analytics

Due to the shape of matrix $A = [a_{ij}]$ shape is 4×3, and the shape of matrix $AB = [a_{ik}]$ is 4×2, we can use the matrix multiplication equation to solve for B (we can also infer that matrix B is 3 rows and 2 columns).

To solve for B, we have the matrix multiplication equation:

$$\begin{pmatrix} 5 & 2 & 3 \\ -2 & 3 & 1 \\ 0 & 2 & 2 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} = \begin{pmatrix} 8 & 6 \\ -1 & 9 \\ 2 & 6 \\ 2 & 0 \end{pmatrix}$$

Equating the corresponding elements with the given matrix AB, we get the following system of equations $\Rightarrow \begin{cases} 5b_{11} + 2b_{21} + 3b_{31} = 8 \\ 5b_{12} + 2b_{22} + 3b_{32} = 6 \\ -2b_{11} + 3b_{21} + b_{31} = -1 \\ -2b_{12} + 3b_{22} + b_{32} = 9 \\ 2b_{21} + 2b_{31} = 2 \\ 2b_{22} + 2b_{32} = 6 \\ b_{11} + b_{31} = 2 \\ b_{12} + b_{32} = 0 \end{cases}$ Then solve the equations and get $B = \begin{pmatrix} 1 & 0 \\ 0 & 3 \\ 1 & 0 \end{pmatrix}$.

2 Question 2

2.1 Find Linear Transformation

Consider the vectors $\vec{a} = \begin{pmatrix} 7 \\ 1 \\ 2 \end{pmatrix}$, $\vec{b} = \begin{pmatrix} 8 \\ 1 \\ 4 \end{pmatrix}$, $\vec{c} = \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix}$. Is there a linear transformation f: $\mathbb{R}^3 \to \mathbb{R}^4$ such that the following conditions are satisfied? $f(\vec{a}) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $f(\vec{b}) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$ and $f(\vec{c}) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$. If there is such an f, then find its matrix. If there is not, then explain why.

2.1.1 Analytics

We can represent the matrix f as:

$$f = \begin{pmatrix} | & | & | \\ \vec{f}(\vec{a}) & \vec{f}(\vec{b}) & \vec{f}(\vec{c}) \\ | & | & | \end{pmatrix}$$

Then, set up the following system of equations:

$$\begin{pmatrix} | & | & | \\ \vec{f}(\vec{a}) & \vec{f}(\vec{b}) & \vec{f}(\vec{c}) \\ | & | & | \end{pmatrix} \begin{pmatrix} 7 & 8 & -1 \\ 1 & 1 & 0 \\ 2 & 4 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

we can find the matrix f by solving this system of equations.

```python
import numpy as np

# Define the given vectors
a = np.array([7, 1, 2])
b = np.array([8, 1, 4])
c = np.array([-1, 0, 2])

# Define the target matrices
fa_target = np.array([1, 0, 0, 0])
fb_target = np.array([0, 1, 0, 0])
```

```
fc_target = np.array([0, 0, 0, 0])

# Form the matrix equation [f] * [a, b, c] = [fa_target, fb_target, fc_target]
A = np.vstack((a, b, c)).T  # Transpose to make it 3x3
B_target = np.vstack((fa_target, fb_target, fc_target)).T  # Transpose to make it 3x4

# Solve for the matrix [f]
try:
    f_matrix, residuals, rank, s = np.linalg.lstsq(A, B_target, rcond=None)
    print("Matrix [f]:")
    print(f_matrix)
except np.linalg.LinAlgError:
    print("No solution exists for the given conditions.")
```

Output Analytics: 'No solution exists for the given conditions.' It implies that there is no matrix f satisfying the specified conditions.

2.2 Another Linear Transformation

Is there a linear transformation $f\colon \mathbb{R}^3 \to \mathbb{R}^4$ such that the following conditions are satisfied? $f(\vec{a}) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$, $f(\vec{b}) = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$ and $f(\vec{c}) = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$.

2.2.1 Analytics

Bring $f(\vec{a})$, $f(\vec{b})$, $f(\vec{c})$ to the Python program we created. The output is still **'No solution exists for the given conditions.'**

3 Question 3

3.1 Find Value for x and y

Consider the linear system of equations

$$\begin{pmatrix} 6 & -4 \\ 8 & -6 \\ 2 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Either find a solution or explain why it does not have any solutions.

3.1.1 Analyics

To determine whether the linear system has a solution, we can check the consistency of the system using the augmented matrix. The augmented matrix for the given system is:

$$\left(\begin{array}{cc|c} 6 & -4 & 1 \\ 8 & -6 & 1 \\ 2 & 0 & 1 \end{array}\right) \xrightarrow{Row2=Row2-\frac{4}{3}\times Row1} \xrightarrow{Row3=Row3-\frac{1}{3}\times Row1} \left(\begin{array}{cc|c} 6 & -4 & 1 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & \frac{4}{3} & \frac{2}{3} \end{array}\right)$$

$$\xrightarrow{Row2\times(-3)} \left(\begin{array}{cc|c} 6 & -4 & 1 \\ 0 & 2 & 1 \\ 0 & \frac{4}{3} & \frac{2}{3} \end{array}\right)$$

$$\xrightarrow{Row3\times(\frac{3}{2})} \left(\begin{array}{cc|c} 6 & -4 & 1 \\ 0 & 2 & 1 \\ 0 & 2 & 1 \end{array}\right)$$

$$\xrightarrow{Row3=Row3-Row2} \left(\begin{array}{cc|c} 6 & -4 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \end{array}\right)$$

$$\Rightarrow \begin{cases} 6x - 4y = 1 \\ 2y = 1 \end{cases} \Rightarrow y = \frac{1}{2},\ x = \frac{1}{2}$$

Conclusion: The system has only one determine solution. $x = \frac{1}{2},\ y = \frac{1}{2}$

3.2 Vector Inclusion in Column Span

Determine whether or not the vector $\begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$ in the range (i.e., column span) of the matrix $\begin{pmatrix} 1 & 2 \\ 0 & 2 \\ 1 & 3 \end{pmatrix}$.

3.2.1 Analytics

We can perform row reduction or use the concept of matrix multiplication. The augmented matrix for the given system is:

$$\left(\begin{array}{cc|c} 1 & 2 & 10 \\ 0 & 2 & 10 \\ 1 & 3 & 10 \end{array}\right)$$

Performing row operations:

$$\xrightarrow{R_1=R_1-R_2} \left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 2 & 10 \\ 0 & 1 & 10 \end{array}\right) \xrightarrow{R_3=R_3-\frac{1}{2}R_2} \left(\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 2 & 10 \\ 0 & 0 & 5 \end{array}\right)$$

The last row indicates that the system is inconsistent (no solution), which means that the vector $\mathbf{v} = \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix}$ is not in the column span of the matrix A. Therefore, the vector cannot be represented as a linear combination of the columns of matrix A.

4 Question 4

Consider a discrete dynamical system: $\begin{pmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{pmatrix} = A \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}$.

$$A = \begin{pmatrix} 1.03552632 & 0.01842105 & -0.16447368 \\ -0.00921053 & 1.10263158 & -0.00921053 \\ -0.13815789 & 0.03947368 & 1.06184211 \end{pmatrix}$$

4.1 Caluclate the Next State

If $(x_0, y_0, Z_0) = (100, 100, 100)$, caluclate (x_1, y_1, z_1) and (x_2, y_2, z_2)

4.1.1 Analytics

```python
import numpy as np

# Given matrix A
A = np.array([
    [1.03552632, 0.01842105, -0.16447368],
    [-0.00921053, 1.10263158, -0.00921053],
    [-0.13815789, 0.03947368, 1.06184211]
])

# Initial state (x0, y0, z0)
initial_state = np.array([100, 100, 100])

# Calculate the next states (x1, y1, z1) and (x2, y2, z2)
state_1 = np.dot(A, initial_state)
state_2 = np.dot(A, state_1)

# Print the results
print("Initial State:", initial_state)
print("Next State (x1, y1, z1):", state_1)
print("Next State (x2, y2, z2):", state_2)
```

4.1.2 Result

Initial State: [100 100 100]
Next State (x1, y1, z1): [88.947369 108.421052 96.31579]
Next State (x2, y2, z2): [78.26315889 117.84210399 94.26315877]

4.2 Combination of Eigenvectors

Given the initial state vector $(x_0, y_0, z_0) = (100, 100, 100)$, write this vector as a linear combination of eigenvectors of A.

4.2.1 Analytics

To express the initial state vector $(x_0, y_0, z_0) = (100, 100, 100)$ as a linear combination of the eigenvectors of matrix A, we need to find the eigenvectors and eigenvalues of A.

Let v_1, v_2, v_3 be the eigenvectors of A corresponding to the eigenvalues $\lambda_1, \lambda_2, \lambda_3$, respectively.

The expression for the initial state vector as a linear combination of eigenvectors is given by:

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3$$

where c_1, c_2, c_3 are the coefficients to be determined.

Then, calculate the eigenvectors and eigenvalues using NumPy:

```python
import numpy as np

# Given matrix A
A = np.array([
    [1.03552632, 0.01842105, -0.16447368],
    [-0.00921053, 1.10263158, -0.00921053],
    [-0.13815789, 0.03947368, 1.06184211]
])

# Calculate eigenvectors and eigenvalues
eigenvalues, eigenvectors = np.linalg.eig(A)

# Given initial state vector
x0 = np.array([100, 100, 100])

# Solve for coefficients c1, c2, c3
coefficients = np.linalg.solve(eigenvectors, x0)

```

```
# Print the results
print("Eigenvectors:")
print(eigenvectors)
print("\nEigenvalues:")
print(eigenvalues)
print("\nInitial state vector as a linear combination of eigenvectors:")
print(f"x0 = {coefficients[0]:.2f} * v1 + {coefficients[1]:.2f} * v2 + {coefficients[2]:.2f} * v3")
```

This code calculates the eigenvectors and eigenvalues of matrix A and then solves for the coefficients c_1, c_2, c_3 in the expression $\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3$.

Eigenvectors: $\begin{bmatrix} 7.66651880e-01 & 7.07106781e-01 & -2.74721110e-01 \\ 6.38876847e-02 & 4.18086088e-16 & -9.61523953e-01 \\ 6.38876559e-01 & -7.07106781e-01 & 9.94056696e-10 \end{bmatrix}$

Eigenvalues: $\begin{bmatrix} 0.90000001 & 1.2 & 1. \end{bmatrix}$

Initial state vector as a linear combination of eigenvectors:

$$x_0 = 123.57v_1 - 29.77v_2 - 95.79v_3$$

4.3 Finding Initial State for Decreasing Norm

Find an initial state vector (x_0, y_0, z_0) of norm 10 such that the norm of (x_i, y_i, z_i) get smaller and smaller as $i \to \infty$.

The general form of the solution for a discrete dynamical system $\mathbf{v}_{i+1} = A\mathbf{v}_i$ is

$$\mathbf{v}_i = c_1\lambda_1^i\mathbf{v}_1 + c_2\lambda_2^i\mathbf{v}_2 + c_3\lambda_3^i\mathbf{v}_3$$

where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of A and $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are the corresponding eigenvectors.

To achieve a decreasing norm as $i \to \infty$, we need to ensure that the eigenvalues have absolute values less than 1. If $|\lambda_i| < 1$, then $\lim_{i\to\infty} \lambda_i^i = 0$, and the norm of $\mathbf{v}_i$ will decrease.

```python
import numpy as np
import matplotlib.pyplot as plt

# Given matrix A
A = np.array([
    [1.03552632, 0.01842105, -0.16447368],
    [-0.00921053, 1.10263158, -0.00921053],
    [-0.13815789, 0.03947368, 1.06184211]
])

# Calculate eigenvectors and eigenvalues
eigenvalues, eigenvectors = np.linalg.eig(A)

# Find eigenvector corresponding to the eigenvalue with the maximum absolute value
max_abs_eigenvalue_index = np.argmax(np.abs(eigenvalues))
initial_eigenvector = eigenvectors[:, max_abs_eigenvalue_index]

# Normalize the eigenvector to have a norm of 10
initial_state = 10 * (initial_eigenvector / np.linalg.norm(initial_eigenvector))

# Initialize lists to store norms and iterations
norms = []
iterations = range(50)  # You can adjust the number of iterations

# Calculate norms for each iteration
for i in iterations:
    current_state = np.linalg.matrix_power(A, i) @ initial_state
    norm_i = np.linalg.norm(current_state)
    norms.append(norm_i)

# Plotting
plt.plot(iterations, norms, marker='o')
plt.title('Norm of (xi, yi, zi) over Iterations')
plt.xlabel('Iterations')
plt.ylabel('Norm')
plt.grid(True)
plt.show()
```

5 Question 5

Find a matrix that has eigenvalues 3; 4; 5 with corresponding eigenvectors:
$\begin{pmatrix}1\\0\\1\end{pmatrix}$, $\begin{pmatrix}0\\1\\2\end{pmatrix}$, $\begin{pmatrix}0\\1\\10\end{pmatrix}$. Explain how you found this matrix.

5.1 Analytics

Construct the matrix A using these eigenvectors as columns. The matrix A is given by:

$$A = \begin{pmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3\end{pmatrix}$$

```python
import numpy as np

# Given eigenvalues
lambda_1 = 3
lambda_2 = 4
lambda_3 = 5

# Given eigenvectors
v1 = np.array([1, 0, 1])
v2 = np.array([0, 1, 2])
v3 = np.array([0, 1, 10])

# Construct the matrix A
A = np.column_stack((v1, v2, v3))

# Display the matrix A
print("Matrix A:")
print(A)

# Check if A has the correct eigenvalues and eigenvectors
for i in range(3):
    result = np.dot(A, A[:, i])
    print(f"Eigenvalue {i+1}:", result)
```

Eigenvalue 1: $\begin{bmatrix}1 & 1 & 11\end{bmatrix}$

Eigenvalue 2: $\begin{bmatrix}0 & 3 & 22\end{bmatrix}$

Eigenvalue 3: $\begin{bmatrix}0 & 11 & 102\end{bmatrix}$