CSE 3380: Linear Algebra for CSE
University of Texas at Arlington
Spring 2023
Alex Dillhoff

Assignment 4

You must show your work to receive credit.

Topics Covered

- Python Basics
- Change of Basis
- Vector Properties

Python Basics

1. Given the matrix A and vector $\mathbf{b}$, complete the following problems in Python.

   Clearly comment in your code which statements belong to which problem. Your submission will include the code file used to answer the problems. Saved your code as `problem1.py`.

$$A = \begin{bmatrix} 3 & 8 & -5 \\ 3 & -6 & -7 \\ 3 & 4 & 2 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -1 \\ -1 \\ 3 \end{bmatrix}$$

   (a) Compute the reduced echelon form of A and convert the result back to a `numpy` array. You will need `sympy` to compute the reduced echelon form.

   (b) Find the column space of A.

   (c) Solve the matrix equation $A\mathbf{x} = \mathbf{b}$.

   (d) Compute Nul A.

2. Create a function that, given a basis $\mathcal{B}$ and a vector $[\mathbf{x}]_{\mathcal{B}}$, computes the vector $\mathbf{x}$. Use functions from `numpy`, `scipy`, and/or `sympy`. Create a function comment header that summarizes your approach.

   In the same file, compute the the vector $\mathbf{x}$ given $\mathcal{B}$ and $[\mathbf{x}]_{\mathcal{B}}$ below and print the result. Save your code as `problem2.py`.

$$\mathcal{B} = \left\{ \begin{bmatrix} 0 \\ -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -4 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 6 \\ 3 \end{bmatrix} \right\}, [\mathbf{x}]_{\mathcal{B}} = \begin{bmatrix} -2 \\ 6 \\ 1 \end{bmatrix}$$

3. Create a python program that determines the dimension of the following set as well as a basis for it. Write a file comment at the top of the file that explains your approach. Your file should print the result. Save your code as `problem3.py`.

$$\left\{ \begin{bmatrix} 2b+3c \\ a+b-2c \\ 4a+b \\ 3a-b-c \end{bmatrix} : a, b, c \in \mathbb{R} \right\}$$

Change of Basis

4. Let $P = \begin{bmatrix} 4 & -9 & 5 \\ -3 & -1 & 6 \\ 9 & -2 & -6 \end{bmatrix}$, $\mathbf{b}_1 = \begin{bmatrix} 0 \\ -1 \\ 3 \end{bmatrix}$, $\mathbf{b}_2 = \begin{bmatrix} 4 \\ 5 \\ -4 \end{bmatrix}$, and $\mathbf{b}_3 = \begin{bmatrix} 3 \\ 3 \\ -6 \end{bmatrix}$.

   Find a basis $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ for $\mathbb{R}^3$ such that P is the change-of-coordinates matrix from $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ to the basis $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$.

Vector Properties

5. Cosine similarity measures the similarity between two non-zero vectors using the dot product. It is defined as

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}.$$

   A result of -1 indicates the two vectors are exactly opposite, 0 indicates they are orthogonal, and 1 indicates they are the same.

   (a) Write a function in Python that calculates the cosine self-similarity of a set of M vectors of dimension N. The input to the function will be an $M \times N$ matrix. The output will be an $M \times M$ matrix of cosine similarity scores.

   (b) Generate a random $M \times N$ matrix and use it as input to your function to test it.

   (c) Create a `matplotlib` plot and use the `matshow` function to display the scores.

   Example Output

   The figure below shows the cosine similarity result of $N = 10$ vectors with dimension $M = 4$.

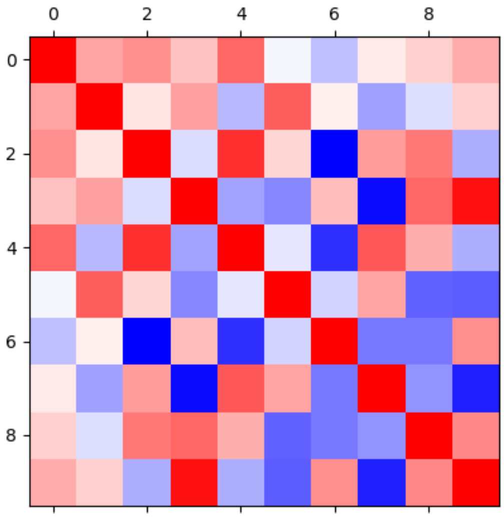

Figure 1: Result of cosine similarity function using the `bwr` color map. Red indicates a similarity score of 1 with pure blue being a similarity of -1.

The Python function and script should be in a single file named `problem5.py`

6. One form of similarity matching requires a database of template samples that is checked against a novel input. The sample in the database most closely matching the input is returned. Using this approach, some information may be gleaned about the input sample even though it has never been observed before.

   In this problem, use the notebook `Hand Pose Search.ipynb` to complete a retrieval system based on 3D keypoints of the hand. Upon implementing the function `find_closest_match`, your solution should accurately find the closest match to the input sample from searching the database. Two similarity scores will be used: L1 and L2 loss.

You may submit your work as either a scanned PDF OR you may take pictures of your homework solutions and combine them into a PDF. Compress the written part with the programming files into a single zip file. **Do not submit individual images. Rename your submission as LASTNAME_ID_A4.zip**.