Math16B – Week 4 Homework

Instructor: Assaf Bar-Natan, Goldsmith 112

Contact: `safibn@brandeis.edu`

Learning Goals

The following are the learning goals for this week. These are a good benchmark for testing your understanding, and these goals will be used to create the evaulations in this course.

Please submit your code as a .py file using the template on LATTE.

Coding Skills

- Break up a large problem into smaller problems
- Create, manipulate, and multiply matrices
- Use the `numpy_linalg` library to find eigenvalues and eigenvectors of matrices

Linear Algebra

- Recall how matrix multiplication works
- Recall the definitions of eigenvectors and eigenvalues

Problems

New: Code documentation: Submissions must be submitted with documentation docstring. Your documentation does not need to conform to any particular standard, but your function names, and descriptions need to be sufficiently clear that anyone who is encountering dictionaries for the first time will be able to understand what you are doing!

For these problems, you may not use the NumPy library (ironic, yet instructional!)

(1) (2 points) Implement matrix-vector multiplication. Define a function `multiply`, whose inputs are a matrix M (given as a list of lists, **NOT** as a NumPy array), and a vector v (given as a list). Your function should return the vector Mv as a list. *Note: You may assume that the dimensions of M and v are appropriate to make Mv well-defined.*

(2) (2 points) Implement the function `transpose`, whose input is a list of lists representing a matrix M, and whose output is a list of lists representing the transposed matrix, M^T.

(3) (2 points) A complex number is a number of the form $z = x + iy$, where $x, y \in \mathbb{R}$ and $i^2 = 1$. We can think of complex numbers as pairs of real numbers: $z \leftrightarrow (x, y)$. Implement a function, `complex_multiply` that multiplies two complex numbers together, and has the following behaviour:
   - Accepts two tuples, $(x_1, y_1), (x_2, y_2)$ representing two complex numbers, z_1 and z_2
   - Returns the tuple corresponding to $z_1 \cdot z_2$.

   Hint: you could do this by hand, but there is a useful trick involving a matrix representation of a complex number. Can you find a matrix M_z such that for any complex number z' (represented as a vector (x', y')), $zz' = M_z(x', y')$?

(4) (2 points) Implement the function `rotate_matrix`, whose input is a **square** matrix M, and whose output is M, but with the entries rotated $90°$ counter-clockwise. For example, if

$$M = \left(\begin{smallmatrix} 1 & 2 \\ 3 & 4 \end{smallmatrix}\right)$$

then `rotate_matrix`(M) is:

$$\left(\begin{smallmatrix} 2 & 4 \\ 1 & 3 \end{smallmatrix}\right)$$

Your solution should work for any size matrix. *Hint: there are many ways to do this, but a nice way is to notice that a rotation is the composition of two reflections.*

(5) (2 points) A square $n \times n$ matrix M is called *positive semidefinite* if for every $x \in \mathbb{R}^n$, $x^T M x \geq 0$. It turns out that a square matrix M is positive semidefinite if and only if:
- M is symmetric, and
- All of the eigenvalues of M are non-negative.

Write a function `is_positive_semidefinite` which checks if a matrix M is positive semidefinite or not. Your function should return True if M is positive semidefinite, and False otherwise. *For this question only, you may use the* `np.linalg.eig` *function.*