Homework 4

IIIT-Delhi
Kaushik Kalyanaraman

Due: November 10, 2019 (Sunday) by 23.59 **Total: 150 points.**

Before you start, please read General Submission Guidelines on Page 2.

Problem 1: Power iteration (30 points)

Implement normalized power iteration to compute the largest magnitude eigenvalue (and corresponding normalized eigenvector) and inverse iteration to compute the smallest magnitude eigenvalue (and corresponding normalized eigenvector) of the matrix:

$$A = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix}$$

Use $x_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ as your starting vector for both methods. You may use a library routine like `numpy.linalg.solve` for solving the linear system in inverse iteration.

Use a general real eigensystem library routine (like `numpy.linalg.eig()` in NumPy) to compute all of the eigenvalues and eigenvectors of the matrix, and compare the results with those obtained by your functions.

Problem 2: Shifted inverse iteration (30 points)

Implement shifted inverse iteration to compute the eigenvalue nearest to 2 and corresponding normalized eigenvector for the matrix:

$$A = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Use an arbitrary starting vector.

Use a real symmetric eigensystem library routine (like `numpy.linalg.eigh()` in NumPy) to compute all of the eigenvalues and eigenvectors of the matrix, and compare the results with those obtained by your function.

Problem 3: Rayleigh quotient iteration (40 points)

Write a function to implement Rayleigh quotient iteration for computing an eigenvalue and corresponding eigenvector of a matrix. Test your program on the matrix in the Problem 3 using an arbitrary starting vector.How does the result compare with the eigenvalues and eigenvector from the library routine you obtained earlier. Finally, assume that the largest magnitude eigenvalue provided by the library routine is the true eigenvalue. Compute the rate of convergence of the Rayleigh quotient iteration to this eigenvalue.

Problem 4: Modified QR iteration (50 points)

Write a function to implement the following version of QR iteration with shifts for computing the eigenvalues of a general real matrix A.

Repeat until convergence:

1. $\sigma = a_{n,n}$ (use corner entry as shift)
2. Compute QR factorization $QR = A - \sigma I$
3. $A = RQ + \sigma I$

Test your function on the matrices in Problems 1 and 2.

General Submission Guidelines

- **This assignment should be answered individually.**
- **You will be penalized for copying or any plagiarism with an automatic zero.**
- The points for each problem is roughly indicative of effort needed for answering that problem. Your mileage may vary!
- IIIT-Delhi academic policies on honesty and integrity apply to all HWs. This includes not copying from one another, from the internet, a book, or any other online or offline source. A repeat offense will be reported to academic administration.
- If you discuss or read secondary sources (other than class notes), please list all your discussion partners and/or secondary sources in your writeup. Failure to do so will constitute violation of honor code.
- All files should be submitted via Google Classroom.
- If your code generates an output figure or table, please provide all such results in a single PDF file along with your code submission.
- You will need to write a separate code for each problem and sometimes for each subproblem as well. You should name each such file as `problem_`n`.py` where n is the problem number. For example, your files could be named `problem_1.py`, `problem_4a.py` and `problem_4b.py` in this HW.
- *Python tip:* You can import Python modules as follows:

```
from __future__ import division
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import numpy.linalg as npla
import scipy.linalg as spla
```

  Every code you write will have one or more of these import statements.