Numerical Analysis 1 – Class 6

Thursday, February 25th, 2021

Subjects covered

- Google's page rank algorithm.
- Power iteration for finding dominant eigenvalue & corresponding eigenvector.
- Inverse iteration and Rayleigh iteration for finding other eigenvalue/vector pairs.
- Simultaneous iteration for finding all eigenvalues/vectors of a matrix.
- QR algorithm for finding multiple eigenvalues
- Principal Component Analysis and relationship to SVD.

Readings

- "The Anatomy of a Large-Scale Hypertextual Web Search Engine", Brin and Page. (Linked on Canvas.) Note that this paper is about Google's page rank system by Google's founders.
- “Iterative Methods for Computing Eigenvalues and Eigenvectors”, Maysum Panju. (Linked on Canvas.)
- “Understanding the QR Algorithm”, D. S. Watkins. (Available on Canvas).
- “A Tutorial on Principal Component Analysis”, J. Shlens. (Linked on Canvas.)

Problems

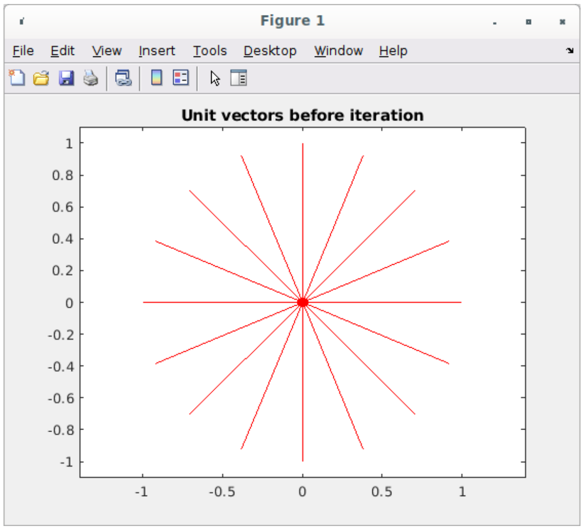
Most of the following problems require you to write a program. For each program you write, please make sure you also write a test which validates your program. Please use Canvas to upload your submissions under the “Assignments” link for this problem set.

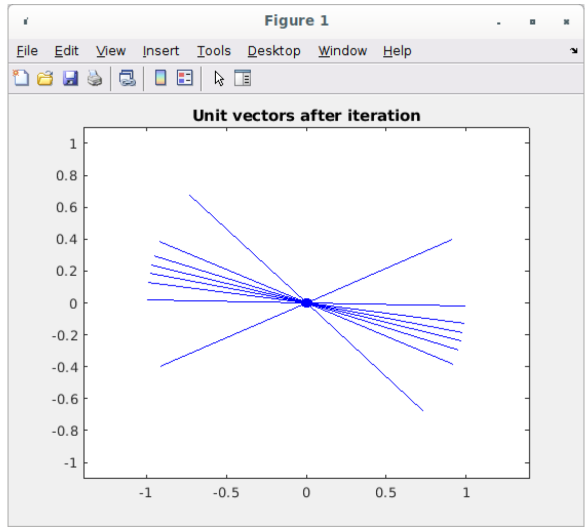
Problem 1

Recall the power method for computing the eigenvector of matrix A corresponding to the largest (absolute value) eigenvalue of A. Please write a program which does the following:

1. Create a set of N unit radius vectors (i.e. vectors which start on the origin and end of the unit circle) equispaced around the unit circle. Then plot the vectors.
2. Accept a 2x2 symmetric matrix A and multiply the vectors by A in a loop. Normalize the vectors after each multiplication, then plot them again. You should see all the vectors collapse into one pair of vectors pointing in opposite directions. A couple of my plots are shown below.
3. After the iteration has converged, please take one of the vectors and use it to compute and print out the associated eigenvalue using the Rayleigh quotient. Also print out the eigenvector itself.
4. To test your program please use Matlab’s eig() function and compare both your eigenvalue and your eigenvector against Matlab’s.

The goal of this exercise is to build a program which allows you to visualize the power iteration method at work.





Problem 2

This problem involves numerically demonstrating some well-known properties of the eigenvalue decomposition. Write a program which generates random matrices *A* of various sizes, performs an eigenvalue decomposition to get its eigenvectors v_j and eigenvalues λ_j , and then checks the following:

- From the definition of eigenvectors and eigenvalues $A v_j = \lambda_j v_j$ please check that v_j and λ_j satisfy $\| A v_j - \lambda_j v_j \| = 0$.
- Please show the expression $A - \lambda_j I$ is singular by computing its condition number.
- Please verify that each eigenvalue, eigenvector pair satisfies the Rayleigh quotient $\lambda_j = v_j^T A v_j / (v_j^T v_j)$.

You may use the Matlab built-in eig() to compute v_j and λ_j .

Problem 3

Back when we discussed the SVD I mentioned the following relationship between an eigenvalue decomposition and the SVD. For any arbitrary matrix A the following holds between the eigenvalues and the singular values:

$$\sqrt{eig(A^T A)} = svd(A)$$

That is,

$$\sqrt{\lambda_j} = \sigma_j \ \ldots \ j = 1, 2, 3, \ldots N$$

where $\lambda_j = eig(A^T A)$ and $\sigma_j = svd(A)$.

Please write a program which generates random matrices A and demonstrates this is true.

Problem 4

In this problem, you will perform PCA on a dataset. I have hidden an object into a larger dataset. The object is saved as a point cloud. Your goal is to write a program implementing PCA to examine the dataset, find its dimensionality, find the principal axes of the data, then do a projection to find the hidden object.

The dataset to process is called "Datafile.csv", available on Blackboard. You may assume the data is organized in rows (as in class). That is, each row is a measurement series, with different samples along the columns. Note that I have added a small amount of noise to all data to simulate the effect of measurement error. Therefore, once you discover the figure, it may have a little bit of noise on it.

You don't need a test function for this problem. Rather, hand in a plot showing the object you found. Also, please answer the following questions:

- How many dimensions of the data hold useful information, and how many are noise?
- What object did I hide in the data?