Numerical Analysis
Assignment 1: Eigenvalues and Eigenvectors

José Daniel Moreno Ceballos,
Daniel Fernando Correa Carreño,
and Mariana Valencia Cubillos

December 2024

Contents

1 Introduction

The problem of finding the eigenvalues and eigenvectors of a matrix A is fundamental in many areas of mathematics, engineering, and computer science. Given a matrix $A \in \mathbb{R}^{n\times n}$, a (real) eigenvalue λ is a scalar such that there exists a non-zero vector $x \in \mathbb{R}^n$ satisfying

$$Ax = \lambda x.$$

In this equation, λ is called an eigenvalue of A, and x is the corresponding eigenvector.

This assignment will cover three methods for computing eigenvalues and eigenvectors of a matrix, each with advantages and limitations. You will implement these methods, analyze their mathematical basis, and test them on specific matrices.

2 Method 1: The Characteristic Polynomial Method

2.1 Theory

We want to determine if there exists a real scalar λ (eigenvalue), or multiple λ's, that satisfy the equation

$$Ax = \lambda x,$$

which can be rewritten as

$$Ax - \lambda x = 0,$$

and further simplified to

$$(A - \lambda I)x = 0,$$

where I is the identity matrix.

This structure resembles the problem of solving a system of linear equations $Mx = b$, where $M \in \mathbb{R}^{n\times n}$. In this context, x is an eigenvector associated with λ if and only if $x \in \text{Nul}(A - \lambda I)$. For this to occur, it is necessary that

$$\det(A - \lambda I) = 0.$$

If instead $\det(A-\lambda I) \neq 0$, then $(A-\lambda I)$ is invertible, implying the existence of $(A-\lambda I)^{-1}$. Applying this inverse to both sides of the equation $(A - \lambda I)x = 0$, we get

$$(A - \lambda I)^{-1}(A - \lambda I)x = (A - \lambda I)^{-1} \cdot 0,$$

which simplifies to $x = 0$. However, we have assumed that x is nonzero.

We conclude that finding the eigenvalues λ corresponds to solving the equation $\det(A-\lambda I) = 0$, which defines the **characteristic polynomial** of the matrix A.

The (real) roots of this polynomial are the eigenvalues of A. To find the eigenvectors, each eigenvalue λ can be substituted back into

$$(A - \lambda I)x = 0$$

to solve the linear equation system for the corresponding eigenvector x. This method requires solving a polynomial equation, which can be computationally challenging for large matrices.

2.2 Error Analysis

This method is considered a direct method, solving $(A - \lambda I)x = 0$ by computing the determinant $\det(A - \lambda I) = 0$ explicitly determines the eigenvalues λ and, consequently, the corresponding eigenvectors. Any errors that arise are solely due to numerical approximations inherent in computer calculations, such as rounding errors or limited precision in representing real numbers.

3 Method 2: The Power Method

3.1 Theory

The Power Method is a numerical algorithm used to approximate the dominant eigenvalue (the eigenvalue with the largest absolute value) and its corresponding eigenvector of a matrix. This method is particularly useful for large matrices where more sophisticated methods may be computationally expensive.

The Power Method starts with a square matrix A of size $n \times n$ (A has some specific conditions we will discuss later) and a arbitrary nonzero vector x_0 it iteratively multiplies x_0 by A and finds the dominant eigenvalue and its corresponding eigenvector.

The procedure is as follows:

1. Choose a starting vector $x^{(0)} \neq 0$, it does not need to be an eigenvector but should not be orthogonal to the dominant eigenvector.
2. We normalize $x^{(0)}$,
$$x^{(0)} = \frac{x^{(0)}}{\|x^{(0)}\|}.$$
3. Now we start iterating, generate a sequence of vectors x_k.
   (a) Our first step is to compute the next vector
   $$y^{(k+1)} = Ax^{(k)}$$
   (b) Then we normalize the vector at each step to prevent overflow or underflow issues
   $$^{(k+1)} = \frac{y^{(k+1)}}{\|y^{(k+1)}\|}$$
   (c) Finally, if the difference between $\|x_{k+1}\|$ and $\|x_k\|$ is smaller than the tolerance we define, then we stop. That is,
   $$\text{If } \|x^{(k+1)} - x^{(k)}\| < \text{tolerance, then stop.}$$
4. After enough iterations we have the dominant eigenvalue λ_1 and its corresponding eigenvector x.
$$\lambda_1 \approx (x^{(k+1)})^\top Ax^{(k+1)}.$$

To understand why the Power Method works, lets explore its mathematical foundation.

Assume that A is an $n \times n$ diagonalizable matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$ and corresponding linearly independent eigenvectors $x_1, x_2, \ldots, x_n$.

Since the eigenvectors span the n-dimensional space, any arbitrary vector $x^{(0)}$ can be expressed as a linear combination of the eigenvectors

$$x = c_1x_1 + c_2x_2 + \cdots + c_nx_n = \sum_{i=1}^{n} c_ix_i.$$

Multiplying both sides of this equation by A, A^2, …, A^k, etc., and recalling that $Ax_i = \lambda_ix_i$, we get

$$\begin{aligned} Ax^{(0)} &= \sum_{i=1}^{n} c_i\lambda_ix_i = y^{(1)}, \\ A^2x^{(0)} &= Ay^{(1)} = \sum_{i=1}^{n} c_i\lambda_i^2x_i = y^{(2)}, \\ &\vdots \\ A^kx^{(0)} &= \sum_{i=1}^{n} c_i\lambda_i^kx_i = y^{(k)}. \end{aligned}$$

Factoring λ_1^k out of the summation, we obtain

$$A^kx = \lambda_1^k \left[c_1x_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^k x_n \right].$$

It doesn't take a lot to notice that $Ax_i = \lambda_ix_i$ will become unbounded, that is why we normalize at each iteration

$$x^{(k)} = \frac{A^kx^{(0)}}{\|A^kx^{(0)}\|}.$$

Since $|\lambda_1| > |\lambda_i|$ for $i = 2, 3, \ldots, n$, the ratios $\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0$ as $k \to \infty$. Therefore,

$$\lim_{k\to\infty} x^{(k)} = \frac{c_1x_1}{\|c_1x_1\|} = \frac{x_1}{\|x_1\|}.$$

So, as $k \to \infty$, the normalized vectors $x^{(k)}$ converge to the dominant eigenvector x_1

Once $x^{(k)}$ approximates x_1, we can estimate the dominant eigenvalue λ_1 using the Rayleigh quotient [LL]

$$\lambda_1 \approx \frac{x^{(k)\top}Ax^{(k)}}{x^{(k)\top}x^{(k)}}.$$

3.2 Error Analysis

3.2.1 Convergence Rate

We can define the error of our vector $x^{(k)}$ at iteration k as

$$e^{(k)} = x^{(k)} - \frac{x_1}{\|x_1\|}.$$

And recalling, we have that

$$x^{(k)} = \frac{A^k x^{(0)}}{\|A^k x^{(0)}\|} = \frac{c_1 x_1 + \sum_{i=2}^n c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i}{\left\| c_1 x_1 + \sum_{i=2}^n c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i \right\|}$$

When $k \to \infty$ the terms involving $\left(\frac{\lambda_i}{\lambda_1}\right)^k$ for $i \geq 2$ become small, and the denominator can be approximated.

$$\left\| c_1 x_1 + \sum_{i=2}^n c_i \left(\frac{\lambda_i}{\lambda_1}\right)^k x_i \right\| \approx \|c_1 x_1\| = |c_1|$$

Thus, we can write the error as

$$e^{(k)} \approx \sum_{i=2}^n \frac{c_i}{|c_1|} \left|\frac{\lambda_i}{\lambda_1}\right|^k x_i$$

Let's consider the norm of this error and use that x_i are normalized. Therefore,

$$\|e^{(k)}\| \leq \sum_{i=2}^n \left|\frac{c_i}{c_1}\right| \left|\frac{\lambda_i}{\lambda_1}\right|^k \|x_i\| = \sum_{i=2}^n \left|\frac{c_i}{c_1}\right| \left|\frac{\lambda_i}{\lambda_1}\right|^k$$

Since the eigenvalues $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$, then we can write

$$\|e^{(k)}\| \leq \left(\sum_{i=2}^n \left|\frac{c_i}{c_1}\right|\right) \left|\frac{\lambda_2}{\lambda_1}\right|^k$$

This shows that the error decreases at a rate proportional to $\left|\frac{\lambda_2}{\lambda_1}\right|^k$. This means that the rate of convergence depends on the magnitude of $\left|\frac{\lambda_2}{\lambda_1}\right|$ thus a smaller ratio results in faster convergence.

3.2.2 Numerical Errors

In addition to the theoretical error analyzed above, numerical errors arising from finite-precision arithmetic must also be considered. These include errors introduced by operations such as normalization, matrix-vector multiplication, and rounding. These numerical errors can accumulate over iterations, potentially impacting the accuracy of the computed eigenvector and eigenvalue.

4 Method 3: QR Decomposition Method

4.1 Theory

Suppose we want to apply the Power Method to n vectors simultaneously. Instead of working with a single vector $x^{(0)}$, we concatenate all our initial guess vectors into a matrix $X^{(0)}$, where each column corresponds to one initial vector. Specifically:

$$X^{(0)} = \begin{bmatrix} x_1^{(0)} & x_2^{(0)} & \cdots & x_n^{(0)} \end{bmatrix}.$$

Issues with This Approach

1. Without additional adjustments, all columns of $X^{(k)}$ will converge to the dominant eigenvector of A.

2. The columns of $X^{(k)}$ can grow or shrink exponentially, leading to numerical overflow or underflow.

To address these issues, we modify our approach. Instead of directly working with $X^{(0)}$, we factor $A^{(k)}$ using the QR decomposition:

$$A^{(k)} = Q^{(k)} R^{(k)},$$

where $Q^{(k)}$ is orthonormal (i.e., ${Q^{(k)}}^T Q^{(k)} = I$) and $R^{(k)}$ is upper triangular. Using this decomposition, we initialize $Q^{(0)}$, ensuring orthogonality and scaling control. This means we will have n vectors pointing in n orthogonal directions, ensuring that all columns of $Q^{(k)}$ have nonzero components in the direction of the desired eigenvalues, and also we would not suffer from underflow or overflow problems.

With $Q^{(0)}$ initialized, we iterate using the following relationship:

$$A^{(k+1)} = {Q^{(k)}}^{-1} A^{(k)} Q^{(k)}.$$

This process is known as Orthogonal Iteration, and it does find all eigenvalues but it has some disadvantages: - If A is dense, it can be expensive to compute the QR decomposition and the inverse (or transpose) of $Q^{(k)}$ in every iteration. - We must ensure that all the columns of $Q^{(k)}$ have nonzero components in the direction of the desired eigenvalue.

To simplify the iteration, we take advantage of the QR decomposition:

$$A^{(k)} = Q^{(k)} R^{(k)}.$$

Substituting this into the orthogonal iteration formula:

$$A^{(k+1)} = {Q^{(k)}}^{-1} A^{(k)} Q^{(k)} = {Q^{(k)}}^{-1} (Q^{(k)} R^{(k)}) Q^{(k)} = R^{(k)} Q^{(k)}.$$

We can iterate with:

$$A^{(k+1)} = R^{(k)} Q^{(k)},$$

and this is what we call the QR Method.

The QR Method iteratively computes the QR decomposition of $A^{(k)}$:

$$A^{(k)} = Q^{(k)} R^{(k)},$$

and updates:

$$A^{(k+1)} = R^{(k)} Q^{(k)}.$$

Over iterations, $A^{(k)}$ converges to an upper triangular matrix whose diagonal entries are the eigenvalues of A.

Since the QR method is a generalization of the Power Method, the convergence of this method relies on the convergence of the latter but also on the capacity of the matrix A to be factored into a QR form. Householder's method is an iterative method that factors A as QR. However, it requires A to be symmetric (or similar to it). Thus, transitively, we will have the same condition over A for the QR method.

4.2 Error Analysis

In the algorithm the criteria of stop it is often used the euclidean norm between the matrix at the current iteration and the previous one. The following theorem summaries the information of the rate of convergence in the case A is symmetric

Theorem Let the QR algorithm be applied to a real symmetric matrix A whose eigenvalues satisfy

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|$$

and whose corresponding eigenvector matrix Q has all non-singular leading principal minors. Then as $k \to \infty$, $A^{(k)}$ converges linearly with constant

$$\max_k \frac{|\lambda_{k+1}|}{|\lambda_k|}$$

to $\operatorname{diag}(\lambda_1, \ldots, \lambda_m)$, and $Q^{(k)}$ (with the signs of its columns adjusted as necessary) converges at the same rate to Q. [LT page 218]

This follows directly from the expression we derived for the error in the Power Method. Note that in the QR method, the matrix Q, being orthogonal, contains a column that aligns with the direction of the desired eigenvector while remaining orthogonal to other eigenvector directions. Since the most dominant eigenvalues converge first (as discussed in Section 3.1.1), this process minimizes the amplification of errors associated with the dominant eigenvalue. Consequently, the maximum error in a given iteration is determined by the largest ratio between the most dominant eigenvalue and the second most dominant eigenvalue that has not yet converged.

5 Conclusions

The methods explored in this assignment offer different approaches to computing eigenvalues and eigenvectors, each with distinct advantages and limitations as described below.

The Characteristic Polynomial Method is best suited for small matrices, where solving the determinant equation $\det(A - \lambda I) = 0$ is straightforward. It provides exact solutions for eigenvalues and eigenvectors, making it a reliable choice for smaller problems. However, as the matrix size increases, this method becomes impractical due to the complexity of solving high-degree polynomials. Additionally, it can be sensitive to numerical errors, particularly when working with matrices that have closely spaced eigenvalues.

The Power Method is ideal for large matrices when only the largest eigenvalue and its corresponding eigenvector are needed. It is simple, efficient, and particularly effective for sparse matrices. However, the is limited to only finding the dominant eigenvalue and may converge slowly if the ratio $|\lambda_2/\lambda_1|$ is close to 1. Furthermore, numerical errors can accumulate over iterations, potentially affecting the accuracy of the results.

The QR Method is specially useful when we need to find all eigenvalues and eigenvectors of a matrix. It is particularly effective for symmetric matrices, where it guarantees convergence under certain conditions. However, its computational cost is significantly higher than the other methods (especially for large dense matrices). Repeated QR factorizations can be slow and require substantial computational resources, making this method less practical for extremely large datasets.

So, how should we go on about choosing one of the three methods? For small matrices, the Characteristic Polynomial Method offers simplicity and accuracy. For large matrices where only the dominant eigenvalue is needed, the Power Method is an efficient choice. When all eigenvalues and eigenvectors are required, and computational resources are available, the QR Method is the most suitable option.

References

[LT] Lloyd N. Trefethen and David Bau III, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[LL] Lap Chi Lau. *CS 860: Advanced Topics in Algorithms - Linear Algebra Notes.* University of Waterloo. Available at: `https://cs.uwaterloo.ca/~lapchi/cs860/notes/02-linear-algebra.pdf`.