Software Assignment

EE24BTECH11066 - YERRA AKHILESH

1 Eigenvalue

1.1 Expression

- Consider an $n \times n$ matrix $\mathbf{A}$ and a nonzero vector $\mathbf{v}$ of length n. If multiplying $\mathbf{A}$ with $\mathbf{v}$(denoted by $\mathbf{Av}$) simply scales $\mathbf{v}$ by a factor of λ, where λ is a scalar, then $\mathbf{v}$ is called an eigenvector of $\mathbf{A}$, and λ is the corresponding eigenvalue. This relationship can be expressed as:

$$\mathbf{Av} = \lambda\mathbf{v}$$

- expressing the above equation into matrix form:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \lambda \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- represented as:

$$(\mathbf{A} - \lambda\mathbf{I})\,\mathbf{v} = \mathbf{0},$$

- above equation has a nonzero solution v if and only if the determinant of the matrix $(A - \lambda I)$ is zero. Therefore, the eigenvalues of A are values of λ that satisfy the equation

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

2 Algorithms

These are some algorithms for finding Eigenvalues of a given matrix:

- QR decomposition
- Power Iteration
- Lanczos Algorithm
- Jacobi Algorithm

3 Chosen Algorithm

3.1 QR decomposition

3.1.1 Introduction

- Main idea of QR decomposition is to repeatedly decompose a matrix $\mathbf{A}$ into product of an orthogonal matrix $\mathbf{Q}\left(\text{i.e.}, \mathbf{Q^TQ} = \mathbf{I}\right)$ and upper triangular matrix $\mathbf{R}$,

  Decompose $\mathbf{A}$ into its QR decomposition:

$$\mathbf{A} = \mathbf{QR}$$

- By applying QR algorithm iteratively, the matrix $\mathbf{A}$ is converted into a diagonal matrix, in which diagonal elements are the required eigenvalues of the given matrix $\mathbf{A}$.

3.1.2 performing QR decomposition

- $\mathbf{A_0}$ is a $n \times n$ matrix, by performing QR decomposition $\mathbf{A_0}$ expressed as:

$$\mathbf{A_0} = \mathbf{Q_0R_0}$$

- form a new matrix $\mathbf{A_1}$ as:

$$\mathbf{A_1} = \mathbf{R_0Q_0}$$

- $\mathbf{A_1}$ is such a matrix that having same eigenvalues as that of $\mathbf{A_0}$.

3.1.3 Repeat QR decomposition

$$\mathbf{A_1} = \mathbf{Q_1R_1}$$

- $\mathbf{A_2}$ can be expressed interms of R_1, Q_1 as

$$\mathbf{A_2} = \mathbf{R_1Q_1}$$

- continue this iterative decomposition until $\mathbf{A_k}$ turns into upper triangular matrix.

3.1.4 Extracting eigenvalues

- Diagonal entrities of $\mathbf{A_k}$ matrix are the required eigenvalues of a matrix $\mathbf{A_0}$ - as the diagonal entrities of a uppertriangular matrix are the eigenvalues of that matrix $\mathbf{A_k}$ is a matrix of same eigenvalues as $\mathbf{A_0}$ as we discussed earlier.

3.2 QR decomposition using Householder Reflections

Algorithm: Given a square matrix $A \in \mathbb{R}^{n\times n}$, the QR decomposition with Householder reflections computes the orthogonal matrix Q and upper triangular matrix R such that

$A = QR$.

Steps:

1) Initialize $Q = I_n$ (identity matrix of size n) and $R = A$.
2) For $k = 1, \ldots, n-1$:
   a) Extract $x = R[k : n, k]$.
   b) Compute the Householder vector v:
   $$v = x + \text{sign}(x_1)\|x\|_2 e_1, \quad v = \frac{v}{\|v\|_2}.$$
   c) Form the Householder matrix:
   $$H = I - 2vv^T.$$
   d) Update R: $R = HR$.
   e) Update Q: $Q = QH^T$.
3) After $n-1$ iterations, Q and R satisfy $A = QR$.

Eigenvalue Iteration:

1) Initialize $A_0 = A$.
2) For $i = 1, 2, \ldots$ (until convergence):
   a) Compute the QR decomposition $Q_i R_i$ of A_{i-1}.
   b) Update $A_i = R_i Q_i$.
3) The diagonal elements of A_i converge to the eigenvalues of A.

4 Time complexity

- Time complexity of QR algorithm is $O\left(n^3\right)$

5 Other insights

5.1 About QR algorithm

- The QR decomposition is a powerful method for computing eigenvalues of a matrix. When used iteratively, the QR algorithm (which is based on QR decomposition) can be helpful to find all eigenvalues of a matrix, even for large matrices, in a stable and efficient manner.

5.2 specific types of matrices

5.2.1 square matrix

- QR decomposition is suitable not only for symmetric matrices but also for non-symmetric matrices.

5.2.2 rectangular matrix

- The QR algorithm is specifically meant for finding eigenvalues of square matrices, so it doesn't directly apply to rectangular ones. However, ideas from QR decomposition are still valuable for working with rectangular matrices, such as in solving least-squares problems or computing singular values. For a more direct approach to analyzing rectangular matrices, methods like singular value decomposition (SVD) or other matrix factorizations are generally used.

5.2.3 special matrix - sparse matrix

- QR decomposition is particularly good at keeping the matrix sparse during computations, especially when using methods like Givens rotations or Householder transformations. This is a big advantage for large, sparse matrices because it helps save both memory and processing power.

5.3 convergence rate

- The QR algorithm has a rapid rate of convergence, specifically quadratic convergence. This means that with each iteration, the accuracy of the eigenvalues improves significantly, often doubling the number of correct digits. For matrices that are well-conditioned, the algorithm usually requires only a relatively small number of iterations to produce accurate results, typically between 10 and 20 even for large matrices.

5.4 Memory usage

- The memory usage of the QR algorithm for finding eigenvalues of $n \times n$ matrix is $O\left(n^2\right)$

6 Comparing the Algorithms

6.1 QR decomposition

6.1.1 Pros

- Works for all square matrices (real, complex, symmetric, non-symmetric)
- Computes all eigenvalues simultaneously.
- Maintains accuracy even for ill-conditioned matrices.

6.1.2 Cons

- Requires multiple iterations to converge.
- The algorithm requires the storage of intermediate matrices and calculations, which can be memory-intensive for large matrices.
- Time complexity

6.2 Power Iteration

- A method to find the largest eigenvalue and corresponding eigenvector of a matrix by repeatedly multiplying it by a vector and normalizing the result.

6.2.1 Pros

- Suitable for finding the largest eigenvalue of a sparse matrix.
- Memory-efficient: Only requires storage for the matrix and a single vector.

6.2.2 Cons

- Only finds the eigenvalue with largest magnitude.
- May require many iterations for matrices with eigenvalues close in magnitude.

6.3 Lanczos Algorithm

- A technique to approximate a few eigenvalues and eigenvectors of a large symmetric matrix efficiently by reducing it to a smaller tridiagonal form.

6.3.1 Pros

- Finds smallest and largest eigenvalues effectively.
- Reduces computation by working on a smaller tridiagonal matrix.

6.3.2 Cons

- It fails for non-symmetric matrices.
- loss of orthogonality among the Lanczos vectors due to finite-precision arithmetic

6.4 Jacobi Method

- A method for finding all eigenvalues and eigenvectors of a symmetric matrix by repeatedly rotating the matrix to zero out off-diagonal elements.

6.4.1 Pros

- It works very effectively for symmetric matrices.

6.4.2 Cons

- Only suitable for symmetric matrices.

Method	Type of Matrix	Main Use	Complexity
Power Iteration	Any matrix	Dominant eigenvalue	$O(n^2)$ per iteration
QR Algorithm	Any matrix	All eigenvalues	$O(n^3)$
Lanczos Algorithm	Sparse, symmetric	Large sparse matrices	$O(n^2)$
Jacobi Algorithm	Symmetric	All eigenvalues	$O(n^3)$

7 CONCLUSION - Why QR!

In the realm of eigenvalue computation, QR decomposition stands as the most reliable and versatile method. It strikes a perfect balance between accuracy, stability, and adaptability, making it suitable for a wide range of applications. Its iterative refinement ensures convergence to precise eigenvalues, delivering results that are both consistent and mathematically rigorous.

While algorithms like Lanczos or the Power method may perform well in specific cases, they often lack the robustness required for broader applications. QR decomposition, on the other hand, excels due to its ability to handle dense, non-symmetric, and complex matrices effectively, making it a cornerstone of numerical computation.

Opting for QR decomposition is more than just a practical decision-it represents a commitment to achieving trustworthy and high-quality results. It sets the benchmark for eigenvalue analysis, proving itself to be an indispensable tool in computational mathematics.

References

[1] G. Strang, *Linear Algebra and Its Applications*, 4th ed., Cengage Learning, 2006.
A comprehensive resource for understanding linear algebra concepts, including eigenvalues and eigenvectors.
[2] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, 1997.
Explains numerical methods, including QR decomposition, for computing eigenvalues.
[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, 2013.
Covers advanced algorithms such as QR decomposition, Power Iteration, and the Lanczos Algorithm.
[4] Eigenvalue and Eigenvector Tutorial,
https://www.mathsisfun.com/algebra/eigenvalue.html.
A beginner-friendly online resource for eigenvalues and eigenvectors.
[5] QR Algorithm Overview,
https://en.wikipedia.org/wiki/QR_algorithm.
Details the QR decomposition process and its applications in eigenvalue computation.