Eigenvalues and Singular Values

Introduction

Eigenvalues and Eigenvectors

The algebraic eigenvalue problem is defined as follows.

Consider the equation: $Ax = \lambda x$

The solution (λ, x) is called an eigenpair of A (or right eigenpair) with the scalar λ called eigenvalue and the vector x called an eigenvector.

We can also consider the equation: $y^T A = \lambda y^T$
The solution (λ, y) is a called a left eigenpair. Left and right eigenvalues are the same; that is if λ is a right eigenvalue, it is also a left eigenvalue. Hence we can define an eigentriple (λ, x, y).

The set of all eigenvalues of A is called the spectrum of A: $\lambda(A)$
The maximum modulus of the eigenvalues is called the spectral radius:
$\rho(A) = \max\{|\lambda| : \lambda \text{ is eigenvalue of } A\}$

Geometric Interpretation

Eigenvectors determine directions that are invariant under multiplication with the matrix; so the effect of multiplying by the matrix is particularly simple:
Only the length of a vector (in that direction) changes.
The relative change in length is the corresponding eigenvalue.

Eigenvalues and eigenvectors decompose complicated behavior of general linear transformation into simpler actions.

If we think of the (vector)space defined by an eigenvector: $\text{range}(x) = \{ax : a \in \mathbb{R}\}$
then this space is invariant under the linear transformation defined by A.
For every vector $y = ax$ we have $Ay = \lambda y = \lambda ax \in \text{range}(x)$
So every element of the space remains in the space after multiplication by A.

This also holds for the space defined by multiple eigenvectors: $Ax_i = \lambda_i x_i$
$\text{span}\{x_1, x_2, x_3\} = \{ax_1 + \beta x_2 + \gamma x_3 : a, \beta, \gamma \in \mathbb{R}\}$
$w = ax_1 + \beta x_2 + \gamma x_3 : Aw = a\lambda_1 x_1 + \beta\lambda_2 x_2 + \gamma\lambda_3 x_3 \in \text{span}\{x_1, x_2, x_3\}$
Hence, the term invariant subspace for such spaces.

Applications

Eigenvalue problems occur in many areas of science and engineering, such as structural analysis, and problems determining the stability of the steady state of a problem defined by a time dependent differential equation (system of equations).

Eigenvalues also play important role in analyzing numerical methods.

We will be mainly concerned with real matrices.
However, both theory and algorithms are applicable to complex matrices.

Notationally, only difference with complex matrices is the use of conjugate transpose, denoted by A^H instead of the usual matrix transpose, A^T.

Examples

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \rightarrow \lambda_1 = 1, x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \lambda_2 = 2, x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \rightarrow \lambda_1 = 1, x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \lambda_2 = 2, x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \rightarrow \lambda_1 = -3, x_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \lambda_2 = -1, x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

$$A = \begin{bmatrix} -4 & 2 \\ 2 & -4 \end{bmatrix} \rightarrow \lambda_1 = -6, x_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \lambda_2 = -2, x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

$$A = \begin{bmatrix} 1 & 4 \\ -4 & 1 \end{bmatrix} \rightarrow \lambda_1 = 1 + 4i, x_1 = \begin{bmatrix} 1 \\ i \end{bmatrix}, \lambda_2 = 1 - 4i, x_2 = \begin{bmatrix} 1 \\ -i \end{bmatrix}.$$

$$A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \rightarrow \lambda_1 = 2, x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \lambda_2 = 2, x_2 = \begin{bmatrix} ? \\ ? \end{bmatrix}.$$

Characteristic Polynomial

The eigenvalues of A are determined by the characteristic polynomial of A.

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

So we're looking for (eigen)values λ such that the matrix $(A - \lambda I)$ is singular:

$\det(A - \lambda I) = 0$ (this is a polynomial in λ)

This polynomial is called the characteristic polynomial of A. The eigenvalues of A are defined to be the roots of its characteristic polynomial.

Since eigenvalues of matrix are roots of its characteristic polynomial, the Fundamental Theorem of Algebra implies that an $n \times n$ matrix A always has n eigenvalues. The eigenvalues, however, need be neither distinct nor real.

Complex eigenvalues of a real matrix must come in complex conjugate pairs.

Multiplicity of eigenvalues

Eigenvalues may be single or multiple (single or multiple roots).
An eigenvalue with multiplicity $k > 1$ has k or fewer independent eigenvectors associated with it. It has at least one associated eigenvector. If it has fewer than k independent eigenvectors we call the eigenvalue (and the matrix) defective.

The multiplicity of an eigenvalue as the (multiple) root of the char. polynomial is called its algebraic multiplicity.
The number of independent eigenvectors associated with an eigenvalue is called its geometric multiplicity.

The geometric multiplicity is smaller than or equal to the algebraic multiplicity.

A matrix that is not defective is called diagonalizable: we have the decomposition

$$A = X\Lambda X^{-1} \Leftrightarrow X^{-1}AX = \Lambda = \mathrm{diag}(\lambda_i)$$

where X contains the eigenvectors (as columns) and Λ contains the eigenvalues.

Jordan form of a matrix

For every matrix $A \in \mathbb{C}^{n \times n}$ there exists a nonsingular matrix X such that

$$X^{-1}AX = \mathrm{diag}(J_1, \cdots, J_q)$$

$$J_i = \begin{bmatrix} \lambda_i & 1 & & \\ & \lambda_i & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_i \end{bmatrix} \text{ and } J_i \in \mathbb{C}^{m_i \times m_i}, \text{ and } m_1 + m_2 + \cdots + m_q = n.$$

Each block has one corresponding eigenvector: q independent eigenvectors
Each block has $m_i - 1$ principal vectors (of grade 2)

If every block is of size 1, the matrix is diagonalizable
Multiple blocks can have the same eigenvalue: $\lambda_i = \lambda_j$
The sum of the sizes of all blocks with the same eigenvalue λ is the algebraic multiplicity of the eigenvalue λ. The number of blocks with the same eigenvalue λ is the geometric multiplicity of the eigenvalue λ.

Examples

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \rightarrow \det\left(\begin{bmatrix} 1-\lambda & 0 \\ 0 & 2-\lambda \end{bmatrix}\right) = 0 \rightarrow (1-\lambda)(2-\lambda) = 0 \rightarrow \lambda_1 = 1,\ \lambda_2 = 2.$$

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix} \rightarrow \begin{vmatrix} 1-\lambda & 1 \\ 0 & 2-\lambda \end{vmatrix} = 0 \rightarrow (1-\lambda)(2-\lambda) = 0 \rightarrow \lambda_1 = 1,\ \lambda_2 = 2.$$

$$A = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix} \rightarrow \begin{vmatrix} -2-\lambda & 1 \\ 1 & -2-\lambda \end{vmatrix} = 0 \rightarrow \lambda^2 + 4\lambda + 3 = 0 \rightarrow \lambda_1 = -3,\ \lambda_2 = -1.$$

$$A = \begin{bmatrix} 1 & 4 \\ -4 & 1 \end{bmatrix} \rightarrow \begin{vmatrix} 1-\lambda & 4 \\ -4 & 1-\lambda \end{vmatrix} = 0 \rightarrow \lambda^2 - 2\lambda + 17 = 0 \rightarrow \lambda_{1,2} = 1 \pm \tfrac{1}{2}\sqrt{4-68} \rightarrow$$
$$\lambda_{1,2} = 1 \pm i\sqrt{16} \rightarrow \lambda_1 = 1 + i4,\ \lambda_2 = 1 - i4$$

$$A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \rightarrow \begin{vmatrix} 2-\lambda & 1 \\ 0 & 2-\lambda \end{vmatrix} = 0 \rightarrow \lambda_1 = 2,\ \lambda_2 = 2.$$

Multiple eigenvalue with algebraic multiplicity 2, but geometric multiplicity 1.

Properties of Eigenvalue Problems

The properties of eigenvalue problems affect the choice of algorithm and software:

- Are all of eigenvalues needed, or only a few?
- Are only eigenvalues needed, or are corresponding eigenvectors also needed?
- Is matrix real or complex?
- Is matrix relatively small and dense, or large and sparse?
- Does matrix have any special properties, such as symmetry, or is it general matrix?

Matrix Properties

Relevant matrix properties:

• Symmetric: $A = A^T, A \in \mathbb{R}^{n \times n}$,	diagonalizable, real eigenvalues, orthogonal eigenvectors
• Hermitian: $A = A^H, A \in \mathbb{C}^{n \times n}$,	diagonalizable, real eigenvalues, orthogonal eigenvectors
• Orthogonal: $A^T A = A A^T = I,\ A \in \mathbb{R}^{n \times n}$,	diagonalizable, unit eigenvalues, orthogonal eigenvectors
• Unitary: $A^H A = A A^H = I,\ A \in \mathbb{C}^{n \times n}$,	diagonalizable, unit eigenvalues, orthogonal eigenvectors
• Normal: $A^H A = A A^H,\ A \in \mathbb{C}^{n \times n}$,	diagonalizable, orthogonal eigenvectors)

Note: $A^H = \bar{A}^T$ sometimes written as $A^* = A^H$

Uniqueness

Eigenvalues and eigenvectors are in general not unique.

We may have multiple eigenvalues (defective or not).

For eigenvectors the case is even more complicated.

- Only direction of eigenvector matters (not length, positive/negative/argument)
- Eigenvectors corresponding to a multiple eigenvalue span invariant subspace of dimension equal to the geometric multiplicity of eigenvalue: any vector in this subspace is an eigenvector.
- Defective eigenvalue still has an associated invariant subspace, but not all vectors in the space are eigenvectors.

Conditioning of eigenproblems

Condition of eigenvalue problem is sensitivity of eigenvalues and -vectors to small perturbations in matrix; not the same as condition number for linear equations.

Different eigenvalues or eigenvectors of given matrix are not necessarily equally sensitive to perturbations in matrix.

Condition of simple eigenvalue: $\kappa(\lambda) = |y^H x|^{-1}$; x and y are corresponding right and left eigenvectors, normalized so that $\|x\| = \|y\| = 1$.

Suppose $(A+E)(x+\Delta x) = (\lambda+\Delta\lambda)(x+\Delta x)$ and drop second order terms:

$$Ax + A\Delta x + Ex = \lambda x + \lambda\Delta x + \Delta\lambda x \quad \Leftrightarrow \quad A\Delta x + Ex = \lambda\Delta x + \Delta\lambda x$$

$$y^H A\Delta x + y^H Ex = \lambda y^H \Delta x + y^H Ex = \lambda y^H \Delta x + \Delta\lambda y^H x$$

$$y^H Ex = \Delta\lambda y^H x \Rightarrow |\Delta\lambda| \le \frac{\|E\|\|y\|\|x\|}{|y^H x|} = \frac{\|E\|}{|y^H x|}$$

Note $\kappa(\lambda)$ gives change in eigenvalue due to (absolute) change in matrix.

Conditioning of eigenproblems

For symmetric or Hermitian matrix, right and left eigenvectors are same, so eigenvalues are inherently well-conditioned. More generally, eigenvalues are well conditioned for normal matrices, but eigenvalues of nonnormal matrices need not be well conditioned.

More generally, if μ is an eigenvalue of $A+E$, then there is a $\lambda \in (A)$:

$$|\mu - \lambda_k| \le \kappa_2(X)\|E\|_2$$

where X is the matrix of eigenvectors of A and κ_2 is its condition number with respect to solving linear systems (and the two-norm).

A useful backward error result is given by the residual.
Let $r = Ax - \lambda x$ and $\|x\|_2 = 1$. Then there exists a perturbation E with $\|E\|_2 = \|r\|_2$ such that $(A+E)x = \lambda x$. Proof: $E = -rx^H$.

Multiple or close eigenvalues can cause numerical difficulties, especially if matrix is defective .

Conditioning of eigenproblems

Eigenvalues of a symmetric (Hermitian) matrix are well-conditioned, but the eigenvectors may not be. As an example consider the following matrix:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1+\varepsilon \end{bmatrix} \rightarrow \lambda_1 = 1,\ x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_2 = 1+\varepsilon,\ x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

If ε is large, it will take a large perturbation to make, for example, the vector $y = [1\ 1]^T$ an eigenvector.

However, if $\varepsilon \to 0$, any vector in the space $\mathrm{span}\{x_1, x_2\}$ becomes an eigenvector with eigenvalue 1. So, a small perturbation $O(\varepsilon)$ will make y an eigenvector. Hence the eigenvectors become ill-conditioned.

For an nonsymmetric matrix the conditioning of the eigenvectors is even more complicated. Apart from close eigenvalues the angles between the eigenvectors play a role and so does the distance to a defective matrix.

Solving Eigenvalues problems

Two steps for systematic methods to compute (all) eigenvalues and eigenvectors

1. For what types of matrices are eigenvalues easily determined
2. What types of transformations preserve eigenvalues

Transformations that preserve eigenvalues are called similarity transformations. Two matrices are called similar if they have the same eigenvalues (not vectors).

Trick is to transform matrices from general form into simple form using similarity transformations (compare with Gaussian elimination for linear systems).

Simple Matrices

Diagonal matrices:
- Eigenvalues are the diagonal entries
- Eigenvectors are the corresponding columns of the identity matrix

Easy, but not every matrix is similar to a diagonal matrix (defective matrices).
Best possible, in general, is bidiagonal with one upper diagonal (Jordan form).

Triangular matrices:
- Eigenvalues are the diagonal entries
- Eigenvectors can be solved for trivially

Always possible; bidiagonal with upper diagonal is upper triangular

Assume simple eigenvalue λ: $(A-\lambda I)x=0$

triangular: $\begin{bmatrix} U_{11} & u & U_{13} \\ 0 & v^T & \\ & & U_{33} \end{bmatrix}$, we set $x=\begin{bmatrix} y \\ -1 \\ 0 \end{bmatrix}$ and we solve for $y \to U_{11}y=u$

For a simple eigenvalue this is always solvable.

Similarity Transformations

What kind of transformations leave eigenvalues unchanged:

Let T be a nonsingular matrix, and $B=T^{-1}AT$ then we say B is similar to A:

$By=\lambda y \Rightarrow T^{-1}ATy=\lambda y \Rightarrow ATy=\lambda Ty$

So λ must (also) be an eigenvalue of A and the corresponding eigenvector is Ty.

So A and B share the same eigenvalues.
They do not have the same eigenvectors, but the eigenvectors of A are easily found from those of B and the similarity transform.

So we want to find a similarity transform that transforms A into a diagonal or triangular matrix.We will do this by a sequence of transforms such that the resulting sequence of matrices converges to the desired form.

Why can't we compute the right similarity transformation at once?

Similarity Transformations

Forms attainable by similarity transforms for various matrices:

Matrix A	Matrix T	Matrix B
all eigenvalues distinct	nonsingular	diagonal
real symmetric	orthogonal	real diagonal
complex Hernitian	unitary	real diagonal
normal	unitary	diagonal
general	unitary	triangular
general	nonsingular	Jordan (almost diagonal)

Eigenvalues and Singular Values

Computing a few eigenvalues

Power Method

Power method computes eigenvalues using a simple iteration.

Choose x_0, and iterate: $x_k = Ax_{k-1} = A^k x_0$

What happens? Assume A is diagonalizable: $A = V\Lambda V^{-1}$ and that $|\lambda_1| > |\lambda_j|$
Write x_0 as sum of eigenvectors (decompose x_0 along eigenvectors):

$x_0 = \Sigma_j v_j a_j$ then $x_k = A^k x_0 = \Sigma_j v_j \lambda_j^k a_j \quad = \quad \lambda_1^k (v_1 a_1 + \Sigma_{j\neq 1} v_j \left(\frac{\lambda_j}{\lambda_1}\right)^k a_j)$

So x_k converges to v_1 since relative weight of other eigenvectors goes to zero.

Convergence is determined by the second largest (magnitude) eigenvalue.
Assume $|\lambda_1| > |\lambda_2| \geq |\lambda_j|$, then the rate of convergence is determined by $\frac{|\lambda_2|}{|\lambda_1|}$

So convergence may be very slow if $\frac{|\lambda_2|}{|\lambda_1|} \approx 1$.

The method converges linearly (if it converges).

Power Method

The power method will not converge (to dominant eigenpair) if:

- The starting vector has no component in dominant eigenvector (corresponding to eigenvalue with largest magnitude). Rounding error usually takes care of the problem
- For a real matrix with largest eigenvalue/vector complex
- The matrix has two (or more) distinct eigenvalues of largest magnitude

To avoid overflow or underflow we normalize the vector every iteration:

$$x_k = Ax_{k-1} / \| Ax_{k-1} \|_\infty$$

If assumptions on previous slide are satisfied then

$x_k \to v_1$
$(Ax_{k-1})_j / (x_{k-1})_j \to \lambda_1$ (if j^{th} coefficient nonzero)

Deflation

Suppose we have found the most dominant eigenpair (λ_1, v_1) (largest eigenvalue in magnitude), and we would like to find the second largest eigenvalue and corresponding vector as well(λ_2, v_2).

Find H such that $Hv_1 = ae_1 \Rightarrow HAH^{-1} = \begin{bmatrix} \lambda_1 & b^T \\ 0 & B \end{bmatrix}$

Now B is (n-1)-by-(n-1) matrix with eigenvalues: $\lambda_j, j \neq 1$

Let $By = \lambda_2 y$ then $v_2 = H^{-1}\begin{bmatrix} a \\ y \end{bmatrix}$, where $a = \frac{b^T y}{\lambda_2 - \lambda_1}$

Other approach, let z vector be such that $z^T v_1 = \lambda_1$

Iterate with $A - v_1 z^T$:

- $z = v_1 \lambda_1 \quad (\|v_j\|_2 = 1)$
- $z = y_1 \lambda_1$, where y_1 is corresponding left eigenvector, normalize $y_1^T v_1 = 1$
- $z = A^T e_k$, where $\|v_1\|_\infty = 1$ and $v_{k,1} = 1$.

Inverse Iteration

What if we want the smallest (magnitude) eigenvalue or any other one that is not the largest in magnitude? Inverse iteration:

$x_k = A^{-1}x_{k-1} \quad \Leftrightarrow \quad \text{solve } Ax_k = x_{k-1}$ or
$x_k = (A - sI)^{-1}x_{k-1} \quad \Leftrightarrow \quad \text{solve } (A - sI)x_k = x_{k-1}$

Inverse iteration with A converges to smallest (in magnitude) eigenvalue if unique, and inverse iteration with $(A - sI)$ converges to eigenvalue/vector closest to s.
By choosing s more accurate we can improve convergence speed.

$(A - sI)^{-1}$ has eigenvalues $(\lambda_j - s)^{-1}$ and so, assuming we want to approximate the eigenpair with (simple) eigenvalue λ_k the rate of convergence is determined by

$$\max_{j \neq k} \frac{|\lambda_k - s|}{|\lambda_j - s|}$$

This can be made arbitrarily small if we know λ_k sufficiently accurate.

Rayleigh Quotient Iteration

Let x be an approximate eigenvalue for a real matrix A. Then approximate eigenvalue is given by $n \times 1$ linear least squares problem:

$x\hat{\lambda} \approx Ax \Rightarrow x^T x\hat{\lambda} = x^T Ax \quad \Rightarrow \quad \hat{\lambda} = \frac{x^T Ax}{x^T x}$ (Rayleigh quotient)

Let $Av = \lambda v$ and A be symmetric then $\|x - v\| = O(\varepsilon) \Rightarrow |\hat{\lambda} - \lambda| = O(\varepsilon^2)$

Let $x = v + \varepsilon p$, where $v \perp p$, $\|v\|_2 = \|p\|_2 = 1$

$$\frac{x^T Ax}{x^T x} = \frac{(v+\varepsilon p)^T A(v+\varepsilon p)}{(v+\varepsilon p)^T(v+\varepsilon p)} = \frac{v^T Av + 2\varepsilon p^T Av + \varepsilon^2 p^T Ap}{v^T v + 2\varepsilon p^T v + \varepsilon^2 p^T p} = \frac{\lambda + \varepsilon^2 p^T Ap}{1+\varepsilon^2} \approx$$

$$(\lambda + \varepsilon^2 p^T Ap)(1 - \varepsilon^2) = \lambda + O(\varepsilon^2)$$

Rayleigh Quotient Iteration

So given $O(\varepsilon)$ approximation to v we get $O(\varepsilon^2)$ approximation to λ.

Cunning plan: Improve eigenvalue estimate s every step using Rayleigh quotient.

$$x_k = (A - r_{k-1}I)^{-1}x_{k-1} \quad \Leftrightarrow \quad \text{solve } (A - r_{k-1}I)x_k = x_{k-1} \text{ with } r_{k-1} = \frac{x_{k-1}^T Ax_{k-1}}{x_{k-1}^T x_{k-1}}$$

This is called Rayleigh Quotient Iteration
For symmetrix matrices, close to solution, we have cubic convergence.

Let $x = v_k + \varepsilon p = v_k + \Sigma_{j \neq k} v_j a_j$, where $v \perp p$, $\|v\|_2 = \|p\|_2 = 1$

$x = (A - \hat{\lambda}I)^{-1}x = (\lambda_k - \hat{\lambda})^{-1}v_k + \Sigma_{j \neq 1} v_j a_j (\lambda_j - \hat{\lambda})^{-1}$ normalization gives

$x(\lambda_k - \hat{\lambda}) = v_k + (\lambda_k - \hat{\lambda})\Sigma_{j \neq 1} v_j a_j (\lambda_j - \hat{\lambda})^{-1}$ and the norm of the error now gives

$\|x(\lambda_k - \hat{\lambda}) - v_k\|_2 = \|(\lambda_k - \hat{\lambda})\Sigma_{j \neq 1} v_j a_j (\lambda_j - \hat{\lambda})^{-1}\|_2 \leq \|(\lambda_k - \hat{\lambda})\rho \Sigma_{j \neq 1} v_j a_j\| = O(\varepsilon^3)$

Reducing the Cost of Rayleigh Quotient Iteration

The Rayleigh Quotient iteration

$r_{k-1} = \frac{x_{k-1}^T Ax_{k-1}}{x_{k-1}^T x_{k-1}}$ solve $(A - sI)x_k = x_{k-1}$

requires repeatedly solving a (new) linear system.
We can make this cheaper by first reducing A to tridiagonal or Hessenberg form using unitary similarity transformations: $A' = Q^T AQ$

This gives $T = Q^T AQ$ is tridiagonal if A is symmetric, and $H = Q^T AQ$ is (upper) Hessenberg if A is nonsymmetric.

Solving a linear tridiagonal system (order n) takes only $O(n)$ operations.
Solving an upper Hessenberg system (order n) takes $O(n^2)$ operations, for n Givens rotations and backsubstitution $O(n^2)$.

Reduction to Hessenberg or Tridiagonal Form

First we apply a Householder reflection to zero the coefficients in the first column below the second coefficient, and we multiply with the transpose from the right:

$$Q^TAQ = \begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ & x & x & x & x \\ & x & x & x & x \\ & x & x & x & x \end{bmatrix} Q$$

The important idea is that Q^T does not change the first row, and so Q does not change the first column. Hence the multiplication by Q does not destroy the structure of the matrix created by the first step. Next we apply a Householder reflexion that sets the coefficients in the second column to zero below the third coefficient. Continuing in this way we create a sequence of transformations that create the tridiagonal or upper Hessenberg system:

$T = Q_{n-2}Q_{n-3}\cdots Q_1^TAQ_1\cdots Q_{n-3}Q_{n-2}$ same for $H = Q^TAQ$ with $Q = Q_1\cdots Q_{n-2}$

QR Algorithm

The QR-iteration is by far the most well-known algorithm for computing all eigenvalues and eigenvectors of a general (or symmetric) matrix.

The basic step is very simple: Let $A_0 = Q^TAQ$ (tridiagonal or Hessenberg)

$Q_kR_k = A_{k-1}$; $A_k = R_kQ_k$ $(= Q_k^TA_{k-1}Q_k)$ so A_k and A_{k-1} are similar.

It turns out (analysis later) that the subdiagonal coefficients converge to zero, and hence the iteration converges to a upper triangular or diagonal (symmetric) matrix. In practice, also here, we apply shifts to accelerate the convergence.

$Q_kR_k = A_{k-1} - s_kI$; $A_k = R_kQ_k + s_kI$

At each step we set subdiagonal (and superdiagonal symmetric case) entries to zero if they are sufficiently small. This reduces the problem to two problems of smaller size.

QR Algorithm

We will carry out the analysis for the symmetric case:

We have $A_k = Q_k^TA_{k-1}Q_k = Q_k^T\cdots Q_1^TAQ_1\cdots Q_k \Leftrightarrow Q_1\cdots Q_kA_k = AQ_1\cdots Q_k$

Prove by induction: $A^k = Q_1\cdots Q_kR_k\cdots R_1 = \hat{Q}_k\hat{R}_k$
$A = Q_1R_1$;
$A^k = AA^{k-1} = AQ_1\cdots Q_{k-1}R_{k-1}\cdots R_1 = Q_1\cdots Q_{k-1}A_{k-1}R_{k-1}\cdots R_1 =$
$Q_1\cdots Q_{k-1}Q_kR_kR_{k-1}\cdots R_1 = \hat{Q}_k\hat{R}_k$

This gives
$A^k = \hat{Q}_k\hat{R}_k$ so $\hat{Q}_k$ gives an orthogonal basis for $A^k = A^kI$;
that is, we apply the power method simultaneously to $e_1, \ldots, e_n$

$A_k = \hat{Q}_k^TA\hat{Q}_k$ so on the diagonal $\hat{q}_j^TA\hat{q}_j$ (Rayleigh quotients) and off-diagonal $\hat{q}_i^TA\hat{q}_j$. As the vectors $\hat{q}_j$ converge to eigenvectors, the diagonal coefficients converge to the eigenvalues (quadratically) and the off-diagonals to zero (eigenvectors orthogonal).

QR Algorithm

Let some vector $\hat{q}_k$ converge to the dominant eigenvector; then the orthognality restricts the other vectors to be in the subspace orthogonal to the dominant eigenvector. Hence they only have components in the other eigenvectors. So one of the other vectors will converge to the eigenvector corresponding to the second largest eigenvalue (in magnitude), and so on.
(with some complications if there is no single largest (modulus) eigenvalue.)

This will however only give us linear convergence (just as the power method will).

QR Algorithm

So let's apply the same ideas of Inverse and Rayleigh Quotient Iteration.

$A^k = \hat{Q}_k \hat{R}_k \quad \Leftrightarrow \quad A^{-k} = \hat{R}_k^{-1} \hat{Q}_k^{-1} = \hat{Q}_k \hat{R}_k^{-T}$ (last step since A is symmetric)

Let $P = \begin{bmatrix} & & 1 \\ & 1 & \\ 1 & & \end{bmatrix}$, then $A^{-k}P = \hat{Q}_k P \cdot P\hat{R}_k^{-T}P = QR$

So we also carry out inverse iteration simultaneously on the columns of P.
And $\hat{Q}_k P$ gives an orthogonal basis for $A^{-k}P$.

Now we can improve convergence by carrying out shifts (as shown earlier for the QR algorithm) and get the same convergence as for Rayleigh quotient iteration. As shifts we could take again Rayleigh quotients (diagonals of A_k), however this may not always give convergence. Therefore, in practice, we compute the shifts from two-by-two diagonal blocks (Wilkinson shift).

QR algorithm

PL

systematically / iteratively build Schur similarity transformation s.t.

$Q^* A Q$ upper-triangular

first $Q_0^* A Q_0 \equiv A_0$ tridiagonal / upper Hessenberg

(1) $Q_k R_k = A_{k-1}$, $A_k = R_k Q_k$

(maintains Hess. / tridiag. structure)

→ PP

(2) (1) → $R_k = Q_k^* A_{k-1}$ → $R_k Q_k = Q_k^* A_{k-1} Q_k$

(sim. transf.)

(3) $A_k = Q_k^* A_{k-1} Q_k = Q_k^* Q_{k-1}^* A_{k-2} Q_{k-1} Q_k$

(4) $A_k = Q_k^* Q_{k-1}^* \cdots Q_1^* A_0 Q_1 Q_2 \cdots Q_k$ ⇒

(5) $Q_1 Q_2 \cdots Q_k A_k = A_0 Q_1 Q_2 \cdots Q_k$

(6) or stepwise $A_k = Q_k^* A_{k-1} Q_k$ →

(6) $A_{k-1} Q_k = Q_k A_k$

(7) let $Q_1 Q_2 \cdots Q_k = \hat{Q}_k$

(8) $R_k R_{k-1} \cdots R_1 = \hat{R}_k$

(4) → $A_k = \hat{Q}_k^* A_0 \hat{Q}_k$

We can show $A_0^k = \hat{Q}_k \hat{R}_k$ $(= A^k I)$

(by induction)

$A_0 = Q_1 R_1 = \hat{Q}_1 \hat{R}_1$

$A_0^k = A_0 \cdot A_0^{k-1} = A_0 \hat{Q}_{k-1} \hat{R}_{k-1} =$

$A_0 Q_1 Q_2 \cdots Q_{k-1} R_{k-1} R_{k-2} \cdots R_1$

$A_{i-1} Q_i = Q_i A_i \rightarrow$

$Q_1 A_1 Q_2 \cdots =$

$Q_1 \cdots Q_{k-1} \cdot A_k \cdot R_{k-1} \cdots R_1 =$

$Q_1 \cdots Q_{k-1} Q_k R_k R_{k-1} \cdots R_1 = \hat{Q}_k \hat{R}_k$

So $\hat{Q}_k \hat{R}_k$ QR-decomp. of A_0^k $(A_0^k I)$

→ Applying Power method simultaneously to all columns of identity, then orthogonalize.

Without orthog. all col.s generally converge to eig. vector v_1.

$\hat{q}_1^{(k)} \rightarrow v_1$

$\hat{q}_2^{(k)} \cdots \hat{q}_n^{(k)} \perp \hat{q}_1^{(k)}$ so deflated comp in v_1 removed as $\hat{q}_1^{(k)}$ converges

→ constr to space $\mathcal{Q}$ = span $\{v_2 \cdots v_n\}$

(Herm. case)

Intuitively explains (linear) convergence

(More precise: how does process converge)

$\hat{q}_j^{(k)} \rightarrow v_j \quad (\hat{Q}_k \rightarrow V)$

Then $(A_k)_{ij} = (\hat{q}_j^{(k)})^* A_0 \hat{q}_j^{(k)}$ → Rayleigh quotient

Herm. case off-diagonal coefficients

$(\hat{q}_i^{(k)})^* A_0 \hat{q}_j^{(k)} \to 0$ (orthogonality)

How / How fast does process converge?

Simultaneous Iteration → convergence to invariant subspace

$m > n$

$Q_0^{m \times n}$ (orthonorm. col.s)

$Q_1 R_1 = A Q_0$

$Q_2 R_2 = A Q_1 \to Q_2 = A Q_1 R_2^{-1} = A^2 Q_0 R_1^{-1} R_2^{-1}$

etc (power it. on col.s Q_0)

Let $A = V \Lambda V^H$ where $|d_1| \geq |d_2| \geq \cdots \geq |d_n| > |d_{n+1}| \geq \cdots$

$V = [v_1 \; v_2 \cdots v_n \; v_{n+1} \cdots v_m]$

$V_n = [v_1 \cdots v_n] \quad V_C = [v_{n+1} \cdots v_m]$

$\Lambda_n = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix} \quad \Lambda_C = \begin{pmatrix} d_{n+1} & & \\ & \ddots & \\ & & d_m \end{pmatrix}$

Assume $V_n^* Q_0$ nonsingular

→ init. vec.s have comp. in $[v_1 \cdots v_n]$ s.t. comp. vec.s are independent.

$\hat{Q}_k \hat{R}_k = \overline{Q_k R_k \; R_{k-1} R_{k-2} \cdots} = A Q_{k-1} R_{k-1} R_{k-2} \cdots$

$= A^2 Q_{k-2} R_{k-2} \cdots = A^k Q_0$

QR decomp → $R(Q_k) = R(A^k Q_0)$

$$Q_k \hat{R}_k = A^k Q_0 = V \Lambda^k V^H Q_0 =$$

$$\left(V \begin{pmatrix} \Lambda_m^k \\ 0 \end{pmatrix} V^H + V \begin{pmatrix} 0 \\ \Lambda_c^k \end{pmatrix} V^H \right) Q_0 =$$

$$\{ [v_1 \cdots v_m]^* \Lambda_m^k [v_1 \cdots v_m]^* Q_0 + [v_{m+1} \cdots v_m] \Lambda_c^k [\,]^* Q_0$$

$$= \left(V_m \Lambda_m^k + V_c \Lambda_c^k (V_c^* Q_0)(V_m^* Q_0)^{-1} \right) V_m^* Q_0$$

$V_m \Lambda_m^k$: grows $\sim d_m^k$; $V_c \Lambda_c^k$: grows $\leq d_{m+1}^k$; $(V_c^* Q_0)(V_m^* Q_0)^{-1}$ and $V_m^* Q_0$: fixed and do not change space

So, for $k \to \infty$, relative contribution of the V_c vectors $\to 0$

Normalize: Λ_m^k ↓ $\|\cdot\| \leq 1$

$$\left(V_m + V_c \Lambda_c^k (V_c^* Q_0)(V_m^* Q_0)^{-1} \Lambda_m^{-k} \right) (V_m^* Q_0)$$

V_m: $\|\cdot\| = 1$; V_c: $\|\cdot\| = 1$; Λ_c^k: $\|\cdot\| \leq |d_{m+1}|^k$; $(V_c^* Q_0)$: $\|\cdot\| \leq 1$; $(V_m^* Q_0)^{-1}$: $\|\cdot\| = C$; Λ_m^{-k}: $\|\cdot\| \leq |d_m|^{-k}$; $(V_m^* Q_0)$: $\|\cdot\| = 1$

So, $\| V_c \Lambda_c^k (V_c^* Q_0)(V_m^* Q_0)^{-1} \Lambda_m^{-k} \| \leq C \cdot \left| \frac{d_{m+1}}{d_m} \right|^k$

So, $R(Q_k) \xrightarrow{k} ([v_1 \cdots v_m])$ with rate $C \left| \frac{d_{m+1}}{d_m} \right|^k$

Better, even, consider convergence column wise

(different cases depending on $|d_1| > |d_2| \cdots$ or multiple eigenvalues)

Back to QR

$A_k = \hat{Q}_k^* A_0 \hat{Q}_k$ where $|d_n| > |d_{n+1}|$

$\hat{Q}_k \hat{R}_k = A^k \cdot I$ → consider Q_0 (Simult. Iter):

$$Q_0 = [e_1\ e_2 \cdots e_n]$$

then Q_k from simult iter. → first k columns of $\hat{Q}_k$ →

$[q_1^{(k)} \cdots q_n^{(k)}] \to [v_1 \cdots v_n]P$ (at least blockwise)

exact: $[q_1 \cdots v_1 \cdots v_n\ q_{n+1} \cdots q_m]^* A_0 [\cdots] = \begin{pmatrix} (A_k)_{11} & (A_k)_{12} \\ \boxed{0} & (A_k)_{22} \end{pmatrix}$ (exact / not)

not exact → $\|(A_k)_{21}\| = O\left(\left|\frac{d_{n+1}}{d_n}\right|^k\right)$

by construction $(A_k)_{21} = \begin{bmatrix} 0 \cdots 0 & a_{n+1,n}^{(k)} \\ 0 & 0 \\ & \vdots \\ & 0 \end{bmatrix}$ (A_k upper Hess./tridiagonal)

so $a_{n+1,n}^{(k)} \to 0$ with rate $C\left|\frac{d_{n+1}}{d_n}\right|^k$

(convergence for unshifted QR)

$$\hat{Q}_k \hat{R}_k = (A-\mu_k I)(A-\mu_{k-1} I) \cdots (A-\mu_1 I)$$

Note that $A-\mu I$ has eigenvalues

$$d_1-\mu,\ d_2-\mu,\ \cdots,\ d_n-\mu$$

Let $Av_i = d_i v_i$, then

$$(A-\mu I)v_i = d_i v_i - \mu v_i = (d_i-\mu)v_i$$

So, if $\mu = d_{m+1}-\varepsilon$, then the $n \times n$ leading block converges with rate

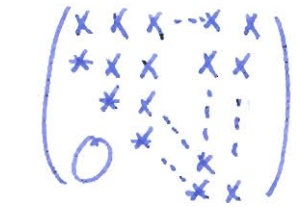
$$O\left(\left|\frac{\varepsilon}{d_m - d_{m+1}+\varepsilon}\right|\right)$$

So, $a_{m+1,m} \to 0$ with this rate.

We can improve convergence.

~

Next we evaluate the effect of these shifts.



* strictly non-zero

Watkins p. 365

A proper upper-Hessenberg matrix (tridiagonal) has all coefficients on the subdiagonal nonzero.

(This is also called unreduced)

(5.6.20) Theo: If $A \in \mathbb{C}^{n \times n}$ is a proper upper Hessenberg (tridiagonal) matrix, then the first $n-1$ columns of A are independent.

Theo 5.6.21 Let $A \in \mathbb{C}^{n \times n}$ be a singular, proper, upper Hessenberg matrix, and let B be the result of one step of the QR algorithm with shift 0 applied to A.

Then the last row of B will be zero. In particular, $b_{nn} = 0$ is an eigenvalue.

proof A singular, but first $n-1$ columns are independent. So, last column dependent with first $n-1$.

$QR = A$, $B = RQ$

We must have $r_{11}, r_{22}, \cdots, r_{n-1\,n-1} \neq 0$, $r_{nn} = 0$.

Since $\text{range}([a_1 \cdots a_{n-1}]) = \text{range}([q_1 \cdots q_{n-1}])$

and $\text{range}([a_1 \cdots a_n]) = \text{range}([a_1 \cdots a_{n-1}])$.

$$R = \begin{pmatrix} x & x & x \\ & x & x \\ & & x \\ 0 & 0 & 0 \end{pmatrix}$$

$$B = RQ = \begin{pmatrix} x & x & x \\ & x & x \\ & & x \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x & x & x \\ x & x & x \\ & x & x \end{pmatrix} = \begin{pmatrix} x & x & x \\ x & x & x \\ & x & x \\ 0 & 0 & 0 \end{pmatrix}$$

Coro. 5.6.23 Let d be an eigenvalue of a proper upper Hessenberg matrix $A \in \mathbb{C}^{n \times n}$. Let B be the result of 1 QR step with shift d applied to A.

Then the last row of B is $[0 \cdots 0\ d]$

and the d eigenvalue can be removed by deflation.

$QR = A - dI$, $B = RQ + dI$

$A - dI$ singular and still proper upper Hess. Hence RQ has last row all zeros $\rightarrow$ B has last row $[0\,0 \cdots 0\ d]$

Duality in subspace iteration
(inverse subspace iteration in disguise)

Let $A \in \mathbb{C}^{n\times n}$, $B = (A^*)^{-1}$, $S \subset \mathbb{R}^n$ a subspace

Define $AS \subset \mathbb{R}^n$ s.t.
$AS = \{Ax : x \in S\}$

Then $(A^m S)^\perp = B^m S^\perp$

$x, y \in \mathbb{C}^n$ and $x \perp y$ (say $x \in S$, $y \in S^\perp$)

$\langle A^m x, B^m y \rangle = y^* (B^*)^m A^m x$

$= y^* A^{-m} A^m x =$

$y^* x = 0$

So, if $x \perp y$ then $A^m x \perp B^m y$

(orthogonality is preserved)

$A^m S \perp B^m S^\perp$

Consider QR algorithm

$A^k = \hat{Q}_k \hat{R}_k$, Let $S = [e_1 \cdots e_m]$
$S^\perp = [e_{m+1}, \cdots, e_n]$

range $(A^k [e_1 \cdots e_m]) = \text{span}\{\hat{q}_1^{(k)} \cdots \hat{q}_m^{(k)}\}$

$\text{span}\{\hat{q}_1^{(k)} \cdots \hat{q}_m^{(k)}\}^\perp = \text{span}\{\hat{q}_{m+1}^{(k)} \cdots \hat{q}_n^{(k)}\}$

$= (A^{-*})^m \text{span}\{e_{m+1} \cdots e_n\}$

So, $[\hat{q}_{m+1}^{(k)} \cdots \hat{q}_n^{(k)}]$ is also result of

inverse iteration with A^*

(same eig. vals as A up to complex conj.)

A Hermitian $\rightarrow (A^*)^{-1} = A^{-1} \rightarrow$ inverse it.

So last column of $\hat{Q}^{(k)}$ results from inverse iteration

(Herm. A) $\quad A^{-k}\,\mathrm{span}\{e_m\} = \mathrm{span}\{\hat{q}_m^{(k)}\}$

So, QR with shifts $(A - \mu I) \longleftrightarrow$

QR with $(A^* - \bar{\mu} I)^{-1} \rightarrow$

quadratic / cubic (Herm. case)

convergence.