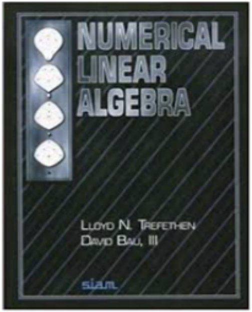
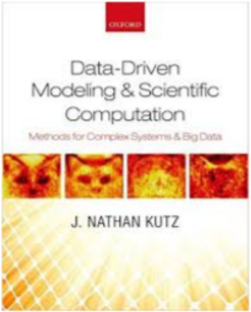
MMAT 5320 Computational Mathematics - Part 1 Numerical Linear Algebra

Andrew Lam

Reference books

- Numerical Linear Algebra by Trefethen and Bau (1997)
- Data-Driven Modeling & Scientific Computation by Kutz (2013)





Topics

Topics: Part 1 – Numerical Linear Algebra by Trefethen and Bau (TB)

- Review of Linear algebra
- Singular value decomposition (SVD)
- QR factorization (Gram–Schmidt/Householder)
- Least squares problem
- Eigenvalue problems
- Eigenvalue algorithms

Topics: Part 2 – Data-Driven Modeling & Scientific Computation by Kutz (K)

- Principal component analysis (PCA)
- Independent component analysis (ICA)
- Compress sensing
- Image denoising and processing
- Data assimilation

§1 - Review of Linear Algebra

How much do you remember?

Let $m, n \in \mathbb{N}$ (natural numbers), x a n-dimensional column vector and A a $m \times n$ matrix:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \left.\underbrace{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}}_{n \text{ columns}}\right\} m \text{ rows}$$

We assume all coefficients $x_1, \dots, x_n$, $a_{11}, \dots, a_{mn}$ are complex numbers, denoted by $\mathbb{C}$, and write

$$x \in \mathbb{C}^n, \quad A \in \mathbb{C}^{m \times n}.$$

Matrix-vector multiplication: The vector $b = Ax$ is the m-dimensional column vector defined as

$$b_i = \sum_{j=1}^{n} a_{ij} x_j \text{ for } i = 1, 2, \dots, m.$$

Matrix-vector multiplication: The vector $b = Ax$ is the m-dimensional column vector defined as

$$b_i = \sum_{j=1}^{n} a_{ij} x_j \text{ for } i = 1, 2, \ldots, m.$$

Graphically: b is a linear combination of the columns of A:

$$\begin{bmatrix} \\ b \\ \\ \end{bmatrix} = \left[\begin{array}{c|c|c|c} & & & \\ a_1 & a_2 & \cdots & a_n \\ & & & \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} \\ a_1 \\ \\ \end{bmatrix} + x_2 \begin{bmatrix} \\ a_2 \\ \\ \end{bmatrix} + \cdots + x_n \begin{bmatrix} \\ a_n \\ \\ \end{bmatrix}$$

"New" way of thinking about matrix-vector products!

Question: Does it makes sense to compute $b = Ax$ where $A \in \mathbb{C}^{n \times m}$ and $x \in \mathbb{C}^n$?

Matrix-matrix multiplication: For $l, m, n \in \mathbb{N}$, and matrices $A \in \mathbb{C}^{l\times m}$ and $C \in \mathbb{C}^{m\times n}$, the product matrix $B = AC$ is a $l \times n$ matrix with entries

$$b_{ij} = \sum_{k=1}^{m} a_{ik} c_{kj} \text{ for } 1 \le i \le l, 1 \le j \le n.$$

Graphically:

$$\left[\begin{array}{c|c|c|c} b_1 & b_2 & \cdots & b_n \end{array}\right] = \left[\begin{array}{c|c|c|c} a_1 & a_2 & \cdots & a_m \end{array}\right] \left[\begin{array}{c|c|c|c} c_1 & c_2 & \cdots & c_n \end{array}\right]$$

column b_1 = linear combination of columns of A with coefficients given by the column c_1

column b_k = linear combination of columns of A with coefficients given by the column c_k

Range and nullspace

A matrix $A \in \mathbb{C}^{m\times n}$ takes a vector $x \in \mathbb{C}^n$ and outputs a vector $b = Ax \in \mathbb{C}^m$.

$$A : \mathbb{C}^n \to \mathbb{C}^m \quad x \mapsto Ax.$$

The range of A (or column space of A), denoted range(A), is the set

$$\{y \in \mathbb{C}^m : y = Ax \text{ for some } x \in \mathbb{C}^n\} \subset \mathbb{C}^m$$

Theorem: range(A) is the space spanned by the columns of A.

The nullspace of A, denoted null(A), is the set

$$\{x \in \mathbb{C}^n : Ax = 0\} \subset \mathbb{C}^n.$$

Example:

- $A \in \mathbb{C}^{n\times n}$ is the identity matrix - range(A) $= \mathbb{C}^n$ and null(A) $= 0$
- $m = 3$, $n = 2$:

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 0 \\ 2 & 0 \end{pmatrix} \quad \text{null}(A) = \left\{ \begin{array}{r} x_1 + 2x_2 = 0 \\ -x_1 = 0 \\ 2x_1 = 0 \end{array} \right\} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}$$

Rank and Inverse

The column rank of a matrix is the dimension of the column space (the number of linearly independent columns). The row rank of a matrix is the dimension of the space spanned by its rows.

Theorem: Row rank = Column rank. $\therefore$ Both are referred to as rank of the matrix.

For a non-square matrix $A \in \mathbb{C}^{m \times n}$, we say A has full rank if

$$\operatorname{rank}(A) = \min(m, n).$$

(What's wrong with taking $\max(m, n)$?)

For a square matrix $A \in \mathbb{C}^{m \times m}$, we say A is invertible/non-singular if $\operatorname{rank}(A) = m$ (i.e., full rank). Then, there is a matrix $Z \in \mathbb{C}^{m \times m}$ (also of full rank) such that

$$AZ = ZA = I.$$

The matrix Z is called the inverse of A, denoted $Z = A^{-1}$.

Keep in mind: In general, $AB \neq BA$ for two matrices $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{n \times m}$.

Adjoint

For a complex number $a = x + iy$, $i = \sqrt{-1}$, its complex conjugate is $\bar{a} = x - iy$. If a is a real number then $a = \bar{a}$.

The hermitian conjugate/adjoint of $A \in \mathbb{C}^{m \times n}$, denoted as A^* is the $n \times m$ matrix with (i, j)th entry

$$a^*_{ij} = \bar{a}_{ji}$$

Graphically:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \quad \Longrightarrow \quad A^* = \begin{bmatrix} \bar{a}_{11} & \bar{a}_{21} & \bar{a}_{31} \\ \bar{a}_{12} & \bar{a}_{22} & \bar{a}_{32} \end{bmatrix}$$

A matrix $A \in \mathbb{C}^{m \times m}$ is hermitian if $A = A^*$.

If $A \in \mathbb{R}^{m \times n}$ is a real matrix, its adjoint is called the transpose, denoted as $A^\top$. If $A \in \mathbb{R}^{m \times m}$ is also hermitian, then A is called symmetric.

Exercise: Show that $(AB)^* = B^* A^*$.

Inner products

The inner product between two vectors $x, y \in \mathbb{C}^m$ is

$$x^* y = \sum_{i=1}^{m} \bar{x}_i y_i.$$

The Euclidean norm of $x \in \mathbb{C}^m$ is

$$\|x\| = \sqrt{x^* x} = \left(\sum_{i=1}^{m} x_i^* x_i \right)^{1/2}$$

and the angle θ between two vectors $x, y \in \mathbb{C}^m$ is

$$\cos\theta = \frac{\mathrm{Re}(x^* y)}{\|x\| \|y\|},$$

where Re denotes the real part.

Properties:

- $\|x\| = \|x^*\|$ and $\|x\| = 0$ if and only if $x = 0$.
- $\theta \in [0, \pi]$.
- $(\alpha x)^*(\beta y) = \bar{\alpha}\beta x^* y$ for $\alpha, \beta \in \mathbb{C}$.

Orthogonality

A pair of vectors x and y are orthogonal if $x^* y = 0$.

Two sets of vectors $X = \{x^1, \dots, x^n\}$ and $Y = \{y^1, \dots, y^m\}$ for $n, m \in \mathbb{N}$ are orthogonal if $(x^i)^*(y^j) = 0$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$.

A set of nonzero vectors S is orthogonal if its elements are pairwise orthogonal, i.e., for all $x, y \in S$ with $x \neq y$, then $x^* y = 0$. [S is orthogonal to itself].

We say S is orthonormal if all elements of S satisfies $\|x\| = 1$.

Theorem: The vectors in an orthogonal set S are linearly independent (LI).

Proof: (1) Suppose to the contrary, $S = \{x^1, \dots, x^n\}$ is not LI. Then, x^n can be written as a linear combination of $\{x^1, \dots, x^{n-1}\}$:

$$x^n = c_1 x^1 + \cdots + c_{n-1} x^{n-1} \quad \text{for } c_i \in \mathbb{C}.$$

(2) Computing $0 < \|x^n\|^2 = (x^n)^*(x^n)$ shows

$$\|x^n\|^2 = (x^n)^* \Big(\sum_{i=1}^{n-1} c_i x^i \Big) = \sum_{i=1}^{n-1} c_i (x^n)^*(x^i) = 0$$

(3) Contradiction.

Decomposition of a vector I

Let $v \in \mathbb{C}^m$ be a vector, and $S = \{q_1, \ldots, q_m\}$ is an orthogonal set. Then, S is a basis of $\mathbb{C}^m$.

But what is the "formula" for v?

Since (scalar) $\times$ vector $=$ vector, if $v = c_1 q_1 + c_2 q_2 + \cdots + c_m q_m$ for scalar $c_i \in \mathbb{C}$, we take the inner product of v with q_k and use orthogonality:

$$q_k^* v = c_1(q_k^* q_1) + c_2(q_k^* q_2) + \cdots + c_n(q_k^* q_n) = c_k(q_k^* q_k) = c_k \|q_k\|^2.$$

So a first formula for v in terms of the set S is

$$v = \sum_{i=1}^{n} \underbrace{\frac{q_i^* v}{\|q_i\|^2}}_{=c_i} q_i \quad \left(\text{or } v = \sum_{i=1}^{n} \underbrace{(q_i^* v)}_{=c_i} q_i \text{ for } S \text{ orthonormal} \right).$$

Decomposition of a vector II

Another viewpoint: we use (matrix) $*$ vector $=$ vector.

For $q, v \in \mathbb{C}^m$, the product $(q^* v)q$ is again a vector in $\mathbb{C}^m$, with jth component

$$[(q^* v)q]_j = \Big(\sum_{i=1}^m \overline{[q]_i}[v]_i\Big)[q]_j = \sum_{i=1}^m [q]_j \overline{[q]_i}[v]_i = \sum_{i=1}^m A_{ji}[v]_i$$

where $A \in \mathbb{C}^{m \times m}$ is the matrix

$$A_{ji} = [q]_j \overline{[q]_i} = (qq^*)_{ji}, \quad A = \begin{pmatrix} [q]_1\overline{[q]_1} & [q]_1\overline{[q]_2} & \dots & [q]_1\overline{[q]_m} \\ [q]_2\overline{[q]_1} & [q]_2\overline{[q]_2} & \dots & [q]_2\overline{[q]_m} \\ \vdots & \vdots & \ddots & \vdots \\ [q]_m\overline{[q]_1} & [q]_m\overline{[q]_2} & \dots & [q]_m\overline{[q]_m} \end{pmatrix}$$

Decomposition of a vector III

Continuing... From the first viewpoint we derive for an orthonormal set S:

$$v = \sum_{i=1}^{m} (q_i^* v) q_i = \sum_{i=1}^{m} (q_i q_i^*) v = \sum_{i=1}^{m} A_i v,$$

where matrices $A_i \in \mathbb{C}^{m \times m}$ are defined as $A_i = (q_i q_i^*)$:

$$A_i = \begin{pmatrix} [q_i]_1 \\ [q_i]_2 \\ \vdots \\ [q_i]_m \end{pmatrix} \begin{pmatrix} \overline{[q_i]_1} & \overline{[q_i]_2} & \dots & \overline{[q_i]_m} \end{pmatrix}$$

Summary: Two ways to express a vector v using inner products and orthonormal sets.

(1) v is a sum of coefficients $q_i^* v$ times vectors q_i;

(2) v is the sum of orthogonal projections $(q_i q_i^*)$ of v.

Exercise: What is the rank of the projection matrices A_i?

Unitary matrices I

A matrix $Q \in \mathbb{C}^{m \times m}$ is unitary if $Q^* Q = I$, i.e., $Q^* = Q^{-1}$.

$$\left[\begin{array}{c} q_1^* \\ \hline q_2^* \\ \hline \vdots \\ \hline q_m^* \end{array}\right] \left[\begin{array}{c|c|c|c} q_1 & q_2 & \cdots & q_m \end{array}\right] = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

So,

$$(q_i^* q_j) = \begin{cases} 0 \text{ if } i \neq j, \\ 1 \text{ if } i = j, \end{cases}$$

i.e., the columns $\{q_i\}_{i=1}^m$ of Q forms an orthonormal basis of $\mathbb{C}^m$.

Multiplication: For Q unitary and $x \in \mathbb{C}^m$, the product $Qx \in \mathbb{C}^m$ is the linear combination of columns $\{q_i\}_{i=1}^m$ with coefficient of x:

$$Qx = x_1 q_1 + x_2 q_2 + \cdots + x_m q_m.$$

Unitary matrices II

For Q unitary and $b \in \mathbb{C}^m$, what is the product $Q^* b \in \mathbb{C}^m$?

Expand b in the basis $\{q_i\}_{i=1}^m$:

$$b = (q_1^* b) q_1 + \cdots + (q_m^* b) q_m.$$

Then, using $Q^* q_j = e_j$ (jth standard unit vector), yields

$$Q^* b = (q_1^* b) e_1 + (q_2^* b) e_2 + \cdots + (q_n^* b) e_n = \begin{pmatrix} q_1^* b \\ q_2^* b \\ \vdots \\ q_n^* b \end{pmatrix},$$

i.e., $Q^* b$ is the vector of coefficients of the expansion of b in the basis $\{q_i\}_{i=1}^m$.

Vector norms

A norm $\|\cdot\| : \mathbb{C}^m \to \mathbb{R}$ is a function satisfying

(i) $\|x\| \geq 0$ and $\|x\| = 0$ if and only if $x = 0$.

(ii) $\|\alpha x\| = |\alpha|\|x\|$ for all $\alpha \in \mathbb{C}$.

(iii) $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality).

Examples: for $1 \leq p < \infty$ the p-norm is defined as

$$\|x\|_p = \Big(\sum_{i=1}^{m} |x_i|^p\Big)^{1/p}.$$

For $p = \infty$, the ∞-norm is defined as

$$\|x\|_\infty = \max_{1 \leq i \leq m} |x_i|.$$

Property: For $1 \leq p \leq q \leq \infty$, it holds for any $x \in \mathbb{C}^m$

$$\|x\|_\infty \leq \|x\|_q \leq \|x\|_p \leq \|x\|_1 \leq m\|x\|_\infty.$$

Meaning – all p-norms are equivalent.

Matrix norms

A matrix $A \in \mathbb{C}^{m \times n}$ can be regarded as a vector in $\mathbb{C}^{mn}$, so one example of a norm is the Frobenius norm $\| \cdot \|_F$:

$$\|A\|_F = \Big(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \Big)^{1/2}.$$

Another common choice is the induced matrix norms: let $\| \cdot \|_{(n)}$ and $\| \cdot \|_{(m)}$ be norms on $\mathbb{C}^n$ and $\mathbb{C}^m$. The induced matrix norm $\|A\|_{(m,n)}$ is the smallest number C such that the following holds

$$\|Ax\|_{(n)} \leq C\|x\|_{(m)} \quad \text{for all } x \in \mathbb{C}^m.$$

Equivalently, (exercise)

$$\|A\|_{(m,n)} = \sup_{x \in \mathbb{C}^m, x \neq 0} \frac{\|Ax\|_{(n)}}{\|x\|_{(m)}} = \sup_{x \in \mathbb{C}^m, \|x\|_{(m)}=1} \|Ax\|_{(n)}.$$

Induced p-matrix norms

When $\|\cdot\|_{(n)}$ and $\|\cdot\|_{(m)}$ are taken to be the same p-norm, the induced p-matrix norm for $A \in \mathbb{C}^{m\times n}$ is defined as

$$\|A\|_p := \sup_{x\in\mathbb{C}^m, x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \sup_{x\in\mathbb{C}^m, \|x\|_p=1} \|Ax\|_p.$$

Examples and characterisations:

- the 1-norm, $\|A\|_1$ is the maximum column sum, i.e., $\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^m |a_{ij}|$.
- the 2-norm, $\|A\|_2$ is the square root of the largest eigenvalue of A^*A.
- the ∞-norm, $\|A\|_\infty$ is the maximum row sum, i.e., $\|A\|_\infty = \max_{1\leq i\leq m} \sum_{j=1}^n |a_{ij}|$.

The Frobenius norm $\|\cdot\|_F$ is not an induced matrix norm. But for square matrices $A \in \mathbb{C}^{m\times m}$ it satisfies

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{m}\|A\|_2.$$

Inequalities I

Two positive real numbers (p, q) are said to be conjugate if $\frac{1}{p} + \frac{1}{q} = 1$.

E.g., $(2, 2)$, $(4, \frac{4}{3})$, $(10, \frac{10}{9})$, $(1, \infty)$, etc.

Hölder's inequality for a product of two vectors $x, y \in \mathbb{C}^m$ is

$$|x^* y| = \Big(\sum_{j=1}^{m} \overline{x}_j y_j\Big)^{1/2} \leq \|x\|_p \|y\|_q = \Big(\sum_{j=1}^{m} |x_i|^p\Big)^{1/p} \Big(\sum_{k=1}^{m} |y_k|^q\Big)^{1/q}.$$

The Cauchy–Schwarz inequality is the special case $p = q = 2$:

$$|x^* y| \leq \|x\|_2 \|y\|_2.$$

Lemma: For $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times q}$, it holds for any $1 \leq p \leq \infty$

$$\|AB\|_p \leq \|A\|_p \|B\|_p.$$

Observe:

- this inequality is for matrices;
- note the difference with Hölder's inequality.

Inequalities II

Lemma: For $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times q}$, it holds for any $1 \le p \le \infty$

$$\|AB\|_p \le \|A\|_p \|B\|_p.$$

Proof: (1) Let $x \in \mathbb{C}^n$ be a nonzero vector, and set $y = \frac{x}{\|x\|_p}$. Then, $\|y\|_p = 1$. By property of norm:

$$\|Ay\|_p = \frac{1}{\|x\|_p}\|Ax\|_p.$$

Taking maximum over all such $y \in \mathbb{C}^n$, we see

$$\|A\|_p = \max_{\|y\|_p=1} \|Ay\|_p \ge \frac{\|Ax\|_p}{\|x\|_p} \quad \Rightarrow \quad \|Ax\|_p \le \|A\|_p\|x\|_p.$$

(2) Set $y = Bx$ for $x \in \mathbb{C}^q$ gives

$$\|ABx\|_p = \|Ay\|_p \le \|A\|_p\|y\|_p = \|A\|_p\|Bx\|_p \le \|A\|_p\|B\|_p\|x\|_p.$$

Take maximum over all x such that $\|x\|_p = 1$ gives

$$\|AB\|_p = \max_{\|x\|_p=1} \|ABx\|_p \le \|A\|_p\|B\|_p.$$

§2 - Singular value decomposition (SVD)

Reduced SVD I

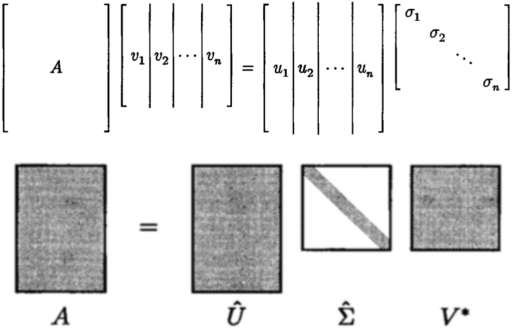
Suppose $A \in \mathbb{C}^{m \times n}$, $m > n$, is a matrix of full rank, i.e., $\text{rank}(A) = n$. We want to find matrices

- $\hat{\Sigma} \in \mathbb{C}^{n \times n}$ diagonal matrix,
- $\hat{U} \in \mathbb{C}^{m \times n}$ with orthonormal columns,
- $V \in \mathbb{C}^{n \times n}$ with orthonormal columns,

such that

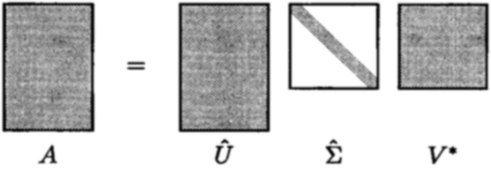
$$AV = \hat{U}\hat{\Sigma} \quad \Leftrightarrow \quad A = \hat{U}\hat{\Sigma}V^* \quad \Leftrightarrow \quad Av_j = \sigma_j u_j$$

Schematically



Reduced SVD II

$$\begin{bmatrix} \\ \\ & A & \\ \\ \\ \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$



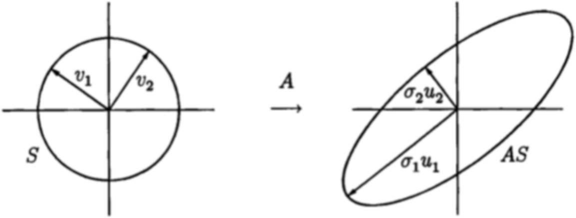
Remarks:

- V is unitary and so V^* is the inverse of V.
- $\hat{U}$ is not unitary since it is not a square matrix.
- Convention: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$ are real numbers, called the singular values of A.
- $\{v_j\}$ are the right singular vectors and $\{u_j\}$ are the left singular vectors.

Geometric viewpoint

Let's take $A \in \mathbb{R}^{m \times n}$ with $m > n$ and full rank.

Geometrically, we can visualise the effects of A on an orthonormal basis $\{v_1, v_2, \ldots, v_n\}$.



E.g., for $n = 2$, $\{v_1, v_2\}$ spans out the unit circle S in $\mathbb{R}^2$. Then, A transforms S to the set AS, which is a (hyper)ellipse in $\mathbb{R}^m$.

Think of taking the unit ball in $\mathbb{R}^m$ and stretching the unit directions $\{e_1, \ldots, e_m\}$ with the vectors $\{\sigma_1 u_1, \ldots, \sigma_m u_m\}$ (the principle semiaxes of the hyperellipise).

The SVD helps us capture the transformation.

Full SVD

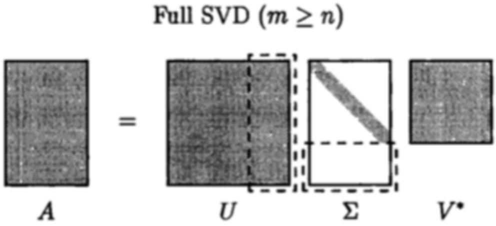
If $A \in \mathbb{C}^{m\times n}$ with $m > n$ is full rank, we have the reduced SVD: $A = \hat{U}\hat{\Sigma}V^*$.

The matrix $\hat{U} \in \mathbb{C}^{m\times n}$ is not unitary, although its columns are orthonormal. So, just add $m-n$ orthonormal columns to $\hat{U}$, leading to a unitary matrix $U \in \mathbb{C}^{m\times m}$. But dimensions don't match now, unless, we add $m-n$ rows of zeros to $\hat{\Sigma}$, leading to a new matrix $\Sigma \in \mathbb{C}^{m\times n}$ (same dim. as A).

The full SVD of a matrix A is

$$A = U\Sigma V^*,$$

where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary, and $\Sigma \in \mathbb{C}^{m\times n}$ has the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ in its diagonal:



Full SVD - non full rank case

If A is not of full rank, i.e., $\text{rank}(A) = p < n$, then the singular values satisfy $\sigma_{p+1} = \cdots = \sigma_n = 0$.

In this case we can only determine $\{u_1, \ldots, u_p\}$ for the left singular vectors and $\{v_1, \ldots, v_p\}$. How do we build U and V?

Simple:

- add $m - p$ (arbitrary) orthonormal rows to the matrix $\tilde{U} = (u_1|u_2|\cdots|u_p)$
- add $n - p$ (arbitrary) orthonormal rows to the matrix $\tilde{V} = (v_1|v_2|\cdots|v_p)$.

Then, the full SVD $A = U\Sigma V^*$ still makes sense.

Theorem: Any matrix $A \in \mathbb{C}^{m\times n}$ admits a (full) singular value decomposition $A = U\Sigma V^*$ with unitary matrices U and V, and a diagonal matrix Σ whose entries are nonnegative real numbers in nonincreasing order.

Proof on the next slide....

Existence proof of full SVD

Proof: (1) The matrix $A^*A \in \mathbb{C}^{n\times n}$ is hermitian and positive semidefinite, i.e.,

$$\overline{(A^*A)^\top} = A^*A, \quad z^*(A^*A)z = \|Az\|_2^2 \geq 0 \text{ for all } z \in \mathbb{C}^n.$$

Hence, the eigenvalues of A^*A are all nonnegative. Let's order them as $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_p^2 > 0$, $\sigma_{p+1}^2 = \sigma_{p+2}^2 = \cdots = \sigma_n^2 = 0$.

(2) Let $\{v_1, \ldots, v_p\}$ be an orthonormal set of eigenvectors for positive eigenvalues, and $\{v_{p+1}, \ldots, v_n\}$ an orthonormal basis for the nullspace of A^*A, i.e., $(A^*A)v_i = \sigma_i^2 v_i$.

(3) We build matrix $V \in \mathbb{C}^{n\times n}$ whose columns are $v_1, \ldots, v_n$, and define for $1 \leq i \leq p$, the vectors $u_i = \frac{1}{\sigma_i}Av_i$. Then, for any $1 \leq i, j \leq p$,

$$u_j^* u_i = \frac{1}{\sigma_i\sigma_j}(Av_j)^*(Av_i) = \frac{1}{\sigma_i\sigma_j}(v_j A^* A v_i) = \frac{\sigma_i}{\sigma_j} v_j^* v_i = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j \end{cases}.$$

So $\{u_1, \ldots, u_p\}$ is an orthonormal set. We add $m - p$ arbitrary orthonormal vectors as columns to build the matrix $U \in \mathbb{C}^{m\times m}$.

(4) Set $\Sigma \in \mathbb{C}^{m\times n}$ to be the diagonal matrix with entries $\sigma_1, \ldots, \sigma_p$ [the positive square root] and zero everywhere else. Then, we claim $A = U\Sigma V^*$.

Proof of claim

Let $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ and $\Sigma \in \mathbb{C}^{m\times n}$ be as above. Then, $AV \in \mathbb{C}^{m\times n}$ and $U\Sigma \in \mathbb{C}^{m\times n}$. Let us compute their ith column:

$$(U\Sigma)_i = \begin{cases} \sigma_i u_i & \text{if } 1 \leq i \leq p, \\ (0,\dots,0)^\top & \text{if } p+1 \leq i \leq n, \end{cases}$$

$$(AV)_i = Av_i = \begin{cases} \sigma_i u_i & \text{if } 1 \leq i \leq p, \\ (0,\dots,0)^\top & \text{if } p+1 \leq i \leq n, \end{cases}$$

since $\{v_{p+1},\dots,v_n\}$ is an orthonormal basis of the nullspace of A^*A, and

$$\text{Nullspace}(A) = \text{Nullspace}(A^*A),$$

which implies $Av_i = 0$ for $i \in \{p+1,\dots,n\}$. $\square$

(Leftover Exercise) Show that for any $A \in \mathbb{C}^{m\times n}$,

$$\text{Nullspace}(A) = \text{Nullspace}(A^*A).$$

Example

Find a SVD for

$$A = \begin{pmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

(1) Compute A^*A:

$$A^*A = \begin{pmatrix} 0 & 0 \\ 0 & 4 \end{pmatrix}$$

and its eigenvalues are $\sigma_1^2 = 4$ and $\sigma_2^2 = 0$ (rank deficient).

(2) Find eigenvectors: $v_1 = (0, 1)^\top$. So we set $v_2 = (1, 0)^\top$. Then, $u_1 = \frac{1}{2}Av_1 = (1, 0, 0)^\top$, and we choose $u_2 = (0, 1, 0)^\top$ and $u_3 = (0, 0, 1)^\top$.

(3) Write down the SVD:

$$\begin{pmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Notice, we could have chosen $u_2 = (0, 0, 1)^\top$ and $u_3 = (0, 1, 0)^\top$, giving a different matrix U and a different SVD. $\therefore$ there can be many SVD for the same matrix.

Rank one representation of SVD

Let $A \in \mathbb{C}^{m\times n}$ and $A = U\Sigma V^*$ be its SVD. Suppose $\operatorname{rank}(A) = r < \min(m,n)$. Then, we can write Σ as a sum of r matrices Σ_j, where $\Sigma_j = \mathrm{diag}(0,\dots,0,\sigma_j,0,\dots,0)$, and

$$A = \sum_{j=1}^{r} U\Sigma_j V^* = \sum_{j=1}^{r} A_j.$$

What does these A_j look like? Let's look at the columns of $U\Sigma_1 \in \mathbb{C}^{m\times n}$. Note that the first column is $\sigma_1 u_1$ and all other columns are zero. Hence,

$$U\Sigma_1 V^* = (\sigma_1 u_1|0|0|\cdots|0)V^* = \sigma_1 \begin{pmatrix} u_1^1\overline{v_1^1} & u_1^1\overline{v_1^2} & \cdots & u_1^1\overline{v_1^n} \\ u_1^2\overline{v_1^1} & u_1^2\overline{v_1^2} & \cdots & u_1^2\overline{v_1^n} \\ \vdots & \vdots & \ddots & \vdots \\ u_1^n\overline{v_1^1} & u_1^n\overline{v_1^2} & \cdots & u_1^n\overline{v_1^n} \end{pmatrix}$$

$$= \sigma_1 u_1 v_1^*.$$

Therefore, the SVD is actually a sum of r rank-one matrices:

$$A = U\Sigma V^* = \sum_{i=1}^{r} \sigma_i u_i v_i^*.$$

Low rank approximation

Let $A \in \mathbb{C}^{m\times n}$ with $\mathrm{rank}(A) = r$, and let $0 \le \nu < r$ be a natural number. We say $A_\nu \in \mathbb{C}^{m\times n}$ is the best rank-ν approximation of A with respect to the norm $\|\cdot\|$ if

$$\|A - A_\nu\| \le \|A - B\| \text{ for all } B \in \mathbb{C}^{m\times n} \text{ s.t. } \mathrm{rank}(B) \le \nu.$$

If

- $\|\cdot\| = \|\cdot\|_2$, the induced 2-norm, then the above inequality is equivalent to

$$\|A - A_\nu\|_2 = \sigma_{\nu+1} \le \|A - B\|_2.$$

- $\|\cdot\| = \|\cdot\|_F$, the Frobenius norm, then the above inequality is equivalent to

$$\|A - A_\nu\|_F = \sqrt{\sigma_{\nu+1}^2 + \cdots + \sigma_r^2} \le \|A - B\|_F.$$

Applications in principle component analysis, total least squares, data compression, etc.

Eckart–Young–Mirsky theorem: In both cases, the answer is simply

$$A_\nu = \sum_{i=1}^{\nu} \sigma_i u_i v_i^*$$

Proof of Eckart–Young–Mirsky theorem (induced 2-norm) I

(1) By definition, for unitary U and V, and diagonal Σ with nonnegative entries in nonincreasing order, the induced 2-norm of $U\Sigma V^*$ is

$$\|U\Sigma V^*\|_2 = \sigma_1.$$

Then,

$$\|A - A_\nu\|_2 = \text{ largest singular value of } (A - A_\nu) = \sigma_{\nu+1}.$$

Note: since $\nu < r$, $\sigma_{\nu+1}$ is always positive!

(2) Let $B \in \mathbb{C}^{m\times n}$ with $\text{rank}(B) = \nu$. Then $\dim \text{Ker}(B) = r - \nu$.

Let $V^{(\nu+1)} = (v_1|\dots|v_{\nu+1}) \in \mathbb{C}^{m\times \nu+1}$, then

$$\dim \text{Ker}(B) + \dim \text{Range}(V^{(\nu+1)}) = r - \nu + \nu + 1 = r + 1.$$

This means there exists a vector $w \in \text{Ker}(B) \cap \text{Range}(V^{(\nu+1)})$, i.e.,

$$w = \gamma_1 v_1 + \dots + \gamma_{\nu+1} v_{\nu+1}.$$

By rescaling γ_i, we can assume $\|w\|_2 = 1$, i.e.,

$$\gamma_1^2 + \dots + \gamma_{\nu+1}^2 = 1.$$

Proof of Eckart–Young–Mirsky theorem (induced 2-norm) II

(3) Recalling the inequality $\|Aw\|_2 \leq \|A\|_2 \|w\|_2 = \|A\|_2$, we see

$$\|A - B\|_2 \geq \|(A - B)w\|_2 = \|Aw\|_2$$

since $w \in \mathrm{Ker}(B)$. But since $AV = U\Sigma$ and

$$w = \gamma_1 v_1 + \cdots + \gamma_{\nu+1} v_{\nu+1}, \text{ with } \gamma_1^2 + \cdots + \gamma_{\nu+1}^2 = 1,$$

it holds

$$Aw = \sum_{i=1}^{\nu+1} \gamma_i A v_i = \sum_{i=1}^{\nu+1} \gamma_i \sigma_i u_i$$

and so

$$\begin{aligned}\|Aw\|_2 &= \Big(\sum_{i=1}^{\nu+1} \gamma_i^2 \sigma_i^2 |u_i|^2 \Big)^{\frac{1}{2}} \geq \sigma_{\nu+1} \Big(\sum_{i=1}^{\nu+1} \gamma_i^2 |u_i|^2 \Big)^{1/2} \\ &= \sigma_{\nu+1} \Big(\sum_{i=1}^{\nu+1} \gamma_i^2 \Big)^{1/2} = \|A - A_\nu\|_2,\end{aligned}$$

where we used $\sigma_i \geq \sigma_{\nu+1}$ for $i = 1, \ldots, \nu$, and $|u_i| = 1$. $\square$

Proof of Eckart–Young–Mirsky theorem (Frobenius norm)

(1) Lemma: Let $A, B \in \mathbb{C}^{m \times n}$ with $\operatorname{rank}(B) \leq k$. Then,

$$\sigma_{i+k}(A) \leq \sigma_i(A - B).$$

$(i + k)$th singular value of A is less than ith singular value of $A - B$.

(2) Recall (from exercise) $\|A\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_p^2}$. Then,

$$\|A - A_\nu\|_F^2 = \sum_{i=\nu+1}^{p} \sigma_i(A)^2 = \sum_{j=1}^{p-\nu} \sigma_{j+\nu}(A)^2$$

(3) Using lemma, we have

$$\|A - A_\nu\|_F^2 \leq \sum_{j=1}^{p-\nu} \sigma_j(A - B)^2 \leq \sum_{j=1}^{p} \sigma_j(A - B)^2 = \|A - B\|_F^2.$$

$\square$

Proof of Lemma

Focus on case $i = 1$, i.e., $\sigma_{k+1}(A) \leq \sigma_1(A+B)$ for $B \in \mathbb{C}^{m\times n}$ with $\mathrm{rank}(B) \leq k$.

(1) Recalling proof of EYM-theorem for induced 2-norm, let $A = U\Sigma V^*$ be the SVD of A, and $V^{(k+1)} = (v_1|\cdots|v_{k+1}) \in \mathbb{C}^{m\times(k+1)}$. Then,

$$\dim \mathrm{Ker}(B) + \dim \mathrm{Range}(V^{(k+1)}) = r + 1,$$

and there exists a vector $w \in \mathrm{Ker}(B) \cap \mathrm{Range}(V^{(k+1)})$.

(2) Looking at $\|Aw\|_2$ and establish lower and upper bounds. First the upper bound:

$$\|Aw\|_2 = \|(A-B)w\|_2 \leq \|A-B\|_2\|w\|_2 = \sigma_1(A-B)\|w\|_2.$$

Now, for the lower bound, let $w = \gamma_1 v_1 + \cdots + \gamma_{k+1}v_{k+1}$, then

$$\|Aw\|_2^2 = \sum_{i=1}^{k+1} \gamma_i^2 \sigma_i(A)^2 \geq \sigma_{k+1}(A)^2 \sum_{i=1}^{k+1} \gamma_i^2 = \sigma_{k+1}(A)^2\|w\|_2^2.$$

Combining:

$$\sigma_{k+1}(A)^2\|w\|_2^2 \leq \|Aw\|_2^2 \leq \sigma_1(A-B)^2\|w\|_2^2.$$

Proof of Lemma (general case) [1]

Now for the general case $\sigma_{i+k}(A) \leq \sigma_i(A-B)$ for $2 \leq i \leq r-k$.

(1) Let $C \in \mathbb{C}^{m\times n}$ with $\text{rank}(C) \leq i+k-1$. Then by previous proof

$$\sigma_{i+k}(A) \leq \sigma_1(A-C).$$

(2) Consider the matrices $(A-B)_{i-1}$ and B_k, where we recall for $A = U\Sigma V^*$, $A_j = \sum_{i=1}^{j} \sigma_i(A) u_i v_i^*$ - the best rank-j approximation of A. Then, the sum $C = (A-B)_{i-1} + B_k$ has at most rank $i+k-1$.

(3) Substitute this matrix C gives

$$\sigma_{i+k}(A) \leq \sigma_1(A-(A-B)_{i-1} - B_k) = \sigma_1((A-B)-(A-B)_{i-1} + (B-B_k))$$

Then, by inequality (exercise)

$$\sigma_1(X+Y) \leq \sigma_1(X) + \sigma_1(Y) \quad \text{for any } X, Y \in \mathbb{C}^{m\times n},$$

we have

$$\begin{aligned}\sigma_{i+k}(A) &\leq \sigma_1((A-B)-(A-B)_{i-1}) + \sigma_1(B-B_k)\\ &= \sigma_i(A-B) + \sigma_{k+1}(B) = \sigma_i(A-B)\end{aligned}$$

as $\text{rank}(B) \leq k$ implies $\sigma_{k+1}(B) = 0$. $\square$

[1] https://www.victorchen.org/2016/01/23/svd-and-low-rank-approximation/

Class exercise

Consider the matrix

$$A = \begin{pmatrix} -2 & 11 \\ -10 & 5 \end{pmatrix}.$$

1. Determine a SVD of A of the form $A = U\Sigma V^*$.
2. List the singular values, left singular vectors, right singular vectors of A, and draw a labelled picture of the unit ball in $\mathbb{R}^2$ and its image under A, together with the singular vectors.
3. What is the induced 1-, 2-, ∞-, and Frobenius norms of A?
4. Use the SVD to compute the inverse of A.
5. What is the best rank-1 approximation A_1 of A with respect to the Frobenius norm? Compute $\|A - A_1\|$ for the 1-, 2-, ∞-, and Frobenius norms.

§3 - QR factorization

Projectors

Definition: A square matrix P is called a projector if

$$P^2 = P.$$

Easy properties:

- If $v \in \text{Range}(P)$, then $Pv = v$.
- For any v, it holds $Pv - v \in \text{Ker}(P)$.
- If P is a projector, so is $I - P$, called the complement, where I is the identity matrix.

Lemma: Let P be a projector. Then,

- $\text{Range}(I - P) = \text{Ker}(P)$, i.e., $I - P$ maps all vectors to $\text{Ker}(P)$.
- $\text{Ker}(I - P) = \text{Range}(P)$,
- $\text{Range}(P) \cap \text{Ker}(P) = \{0\}$, i.e., the projector splits $\mathbb{C}^m$ into two subspaces. Equivalently, for any $w \in \mathbb{C}^m$, there exists (unique) $u \in \mathbb{C}^m$ and $v \in \text{Ker}(P)$ such that $w = Pu + v$.

Exercise: Proof the lemma.

Orthogonal projectors

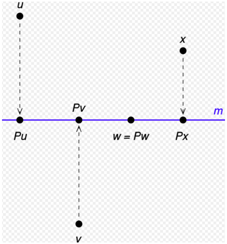
Previous lemma:

$$P \text{ projector} \quad \Rightarrow \quad \mathbb{C}^m = \text{Range}(P) \oplus \text{Ker}(P),$$

where for two subspaces S_1 and S_2, the direct sum $S_1 \oplus S_2$ implies

- $S_1 \cap S_2 = \{0\}$,
- $S_1 + S_2 = \{s = s_1 + s_2 | s_1 \in S_1, s_2 \in S_2\} = \mathbb{C}^m$.

Hence, we say that P is a projection along $\text{Ker}(P)$ onto $\text{Range}(P)$.



E.g., $\text{Ker}(P) = \text{span}\{(0,1)^\top\}$ and $\text{Range}(P) = \text{span}\{(1,0)^\top\}$.

If in addition, $\text{Ker}(P)$ is orthogonal to $\text{Range}(P)$, then we call P a orthogonal projector.

Orthogonal projectors

Theorem: A projector P is orthogonal if and only if P is hermitian, i.e., $P = P^*$.

Proof ($\Leftarrow$): Let P be a hermitian matrix and a projector.

Then, setting $S_1 = \mathrm{Range}(P)$ and $S_2 = \mathrm{Ker}(P)$, we have to show $S_1 \cap S_2 = \{0\}$, $S_1 \oplus S_2 = \mathbb{C}^m$ and $S_1 \perp S_2$.

Since $\mathrm{Range}(I - P) = \mathrm{Ker}(P) = S_2$, the inner product between any two elements $Px \in S_1$ and $(I - P)y \in S_2$ is

$$x^* P^* (I - P) y = x^* (P - P^2) y = x^* (P - P) y = 0,$$

since $P^2 = P$. So $S_1 \perp S_2$ and $S_1 \cap S_2 = \{0\}$.

Furthermore, any $x \in \mathbb{C}^m$ can be written as $x = Px + (I - P)x \in S_1 \oplus S_2$.

Orthogonal projectors

Proof ($\Rightarrow$): Suppose P is an orthogonal projector along S_2 onto S_1 with subspaces $S_1 \oplus S_2 = \mathbb{C}^m$, $S_1 \perp S_2$, and $\dim(S_1) = n < m$.

Let $\{q_1, \dots, q_m\}$ be an orthonormal basis for $\mathbb{C}^m$, where $\{q_1, \dots, q_n\}$ is a basis for S_1 and $\{q_{n+1}, \dots, q_m\}$ a basis for S_2. Then,

$$Pq_j = q_j \text{ for } 1 \leq j \leq n, \text{ while } Pq_j = 0 \text{ for } n+1 \leq j \leq m.$$

To show P is hermitian, we derive the SVD for P. Let $Q \in \mathbb{C}^{m\times m}$ be the unitary matrix with ith column q_i, for $1 \leq i \leq m$. Then,

$$PQ = \left(\begin{array}{c|c|c|c|c} \vdots & & \vdots & \vdots & \\ q_1 & \dots & q_n & 0 & \dots \\ \vdots & & \vdots & \vdots & \end{array}\right)$$

$$Q^*PQ = \begin{pmatrix} I_{n\times n} & 0_{n\times(m-n)} \\ 0_{(m-n)\times n} & 0_{(m-n)\times(m-n)} \end{pmatrix} =: \Sigma.$$

where Σ is a diagonal matrix with 1 in the first n entries. Then, the SVD for P is $P = Q\Sigma Q^*$, and

$$P^* = (Q\Sigma Q^*)^* = Q\Sigma Q^* = P.$$

□

Construction

Recall from slide Decomposition of a vector, if $\{q_1, \ldots, q_m\}$ is an orthonormal basis of $\mathbb{C}^m$, then any vector $v \in \mathbb{C}^m$ can be represented as

$$v = \sum_{i=1}^{m} (q_i q_i^*) v \text{ where } q_i q_i^* \in \mathbb{C}^{m \times m}.$$

Let $\hat{Q} \in \mathbb{C}^{m \times n}$ be the matrix with ith column q_i for $1 \leq i \leq n$.

Claim: $P = \hat{Q}\hat{Q}^*$ is an orthogonal projector onto range$(\hat{Q})$, and

$$Pv = \sum_{i=1}^{n} (q_i q_i^*) v$$

Proof : (1) Easy to check $P = P^*$, and so P is orthogonal projector.

(2) Class Exercise: Show $\hat{Q}\hat{Q}^* = \sum_{i=1}^{n} (q_i q_i^*)$. [Hint: recall the picture]

$$\left[\begin{array}{c|c|c|c} b_1 & b_2 & \cdots & b_n \end{array}\right] = \left[\begin{array}{c|c|c|c} a_1 & a_2 & \cdots & a_m \end{array}\right] \left[\begin{array}{c|c|c|c} c_1 & c_2 & \cdots & c_n \end{array}\right]$$

column b_k = linear combination of columns of A with coefficients given by the column c_k

Construction II

Consequences:

- The projector $P = \sum_{i=1}^{n}(q_i q_i^*) = \hat{Q}\hat{Q}^*$ can be regarded as a sum of rank-one orthogonal projectors:

$$P = \sum_{i=1}^{n} P_i, \quad P_i = q_i q_i^* \in \mathbb{C}^{m \times m}.$$

- The complement $I - P = I - \hat{Q}\hat{Q}^*$ is also an orthogonal projector (onto the space range$(\hat{Q})^{\perp}$) [since $(I-P)^* = (I-P)$].

- For a rank-one projector $A = qq^* \in \mathbb{C}^{m \times m}$ with unit vector q, its complement $A_{\perp} := I - qq^*$ is of rank $m-1$.

Therefore, for an orthonormal basis of $\mathbb{C}^m$, orthogonal projectors can be constructed easily.

But what about if you don't have an orthonormal basis?

Construction III

Suppose $\{a_1, \ldots, a_n\}$ is a set of LI vectors in $\mathbb{C}^m$. How do we construct an orthogonal projector to $\text{span}\{a_1, \ldots, a_n\}$?

Define the matrix $A \in \mathbb{C}^{m\times n}$ whose ith column is a_i for $1 \leq i \leq n$. Then, $\text{range}(A) = \text{span}\{a_1, \ldots, a_n\}$. If $v \in \mathbb{C}^m$ is an arbitrary vector and P is an orthogonal projector onto $\text{range}(A)$ [which we want to identify]. We have

- $y := Pv \in \text{range}(A)$ and $\exists x \in \mathbb{C}^m$ such that $y = Ax$,
- $v - y \in (\text{range}(A))^{\perp}$, and so
- the inner products of a_i and $v - y$ are all zero, i.e., $a_i^*(v - y) = 0$ for $1 \leq i \leq n$.
- In matrix form: $A^*(v - y) = A^*(v - Ax) = 0$ or $A^*Ax = A^*v$.
- Now, $\{a_1, \ldots, a_n\}$ are LI, and so A has full rank, which implies A^*A is invertible.
- Therefore, $x = (A^*A)^{-1}A^*v$ and $y = Ax = A(A^*A)^{-1}A^*v$. Hence,

$$y = Pv = A(A^*A)^{-1}A^*v \quad \Rightarrow \quad P = A(A^*A)^{-1}A^*.$$

If $\{a_1, \ldots, a_n\}$ are orthonormal as well, then $A = \hat{Q}$ and $A^*A = I$, which gives $P = \hat{Q}\hat{Q}^*$ like before.

Example

Consider the matrix

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

What is the orthogonal projector P onto $\mathrm{range}(A) = \{(x, y, x) : x, y \in \mathbb{C}\}$?

(1) Columns of A are LI, so A has full rank.

(2) Matrix calculations:

$$A^*A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}, \quad (A^*A)^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1 \end{pmatrix}$$

$$P = A(A^*A)^{-1}A^* = \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix}$$

Then, the projection of a point $(1, 2, 3)$ to $\mathrm{range}(A)$ is $P(1, 2, 3) = (2, 2, 2)$.

Class exercise

Find the orthogonal projector P onto range(A) for the following matrices

1. $$A = \begin{pmatrix} i & 0 \\ i & 2 \\ -1 & 1-i \end{pmatrix}$$

2. $$A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 3 \\ 2 & 0 & 0 \end{pmatrix}$$

3. $$A = \begin{pmatrix} 0 & 1 & 1 \\ 2 & -1 & 0 \\ 3 & 0 & 0 \end{pmatrix}$$

Matrix factorizations

Motivation: Let $A \in \mathbb{R}^{m \times m}$ where m is a large number, e.g., $m = 10^{90}$. Given $b \in \mathbb{R}^m$, solve $Ax = b$.

If A has a special structure, e.g., diagonal/triangular. The computation $x = A^{-1}b$ can be done (with some effort, but not impossible!)

If A has no such structure, then it is nearly impossible (for us or for computers) to calculate $x = A^{-1}b$.

If A exhibits a factorization, e.g., the Cholesky factorization $A = U^\top U$ where U is upper triangular, then we can find the solution x in two steps:

1. Solve for $U^\top y = b$ or $y = U^{-\top} b$,
2. Then solve for $Ux = y$ or $x = U^{-1}y$.

For non-square matrices $A \in \mathbb{C}^{m \times n}$, the QR factorization splits A into the product of unitary $Q \in \mathbb{C}^{m \times m}$ and upper triangular $R \in \mathbb{C}^{m \times n}$. Then,

$$Ax = b \quad \Leftrightarrow \quad Rx = Q^\top b,$$

which is easier to solve.

Reduced QR factorization

Let $A \in \mathbb{C}^{m \times n}$ be of full rank with columns $a_1, \dots, a_n$. Aim: to find vectors $q_1, q_2, \dots$ such that for each $j = 1, \dots, n$,

$$\text{span}\{q_1, \dots, q_j\} = \text{span}\{a_1, \dots, a_j\}.$$

This is equivalent to

$$\left[\begin{array}{c|c|c|c} & & & \\ a_1 & a_2 & \cdots & a_n \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ q_1 & q_2 & \cdots & q_n \\ & & & \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & \vdots \\ & & \ddots & \\ & & & r_{nn} \end{bmatrix}$$

Why? Recall,

column a_k = linear combination of columns of Q with coefficients given by the column r_k.

$a_1 = r_{11} q_1$ and so if $r_{11} \neq 0$, $\text{span}\{a_1\} = \text{span}\{q_1\}$.
$a_2 = r_{12} q_1 + r_{22} q_2$, and so $a_2 \in \text{span}\{q_1, q_2\}$, which implies

$$\text{span}\{a_1, a_2\} \subset \text{span}\{q_1, q_2\}.$$

For the converse, we see $r_{22} q_2 = a_2 - \frac{r_{12}}{r_{11}} a_1$, so if $r_{22} \neq 0$, we have $q_2 \in \text{span}\{a_1, a_2\}$.

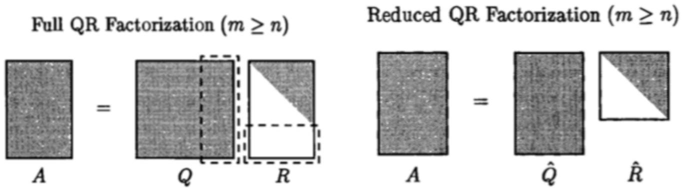
Reduced and Full QR factorization

So, in order for $\text{span}\{a_1, \ldots, a_j\} = \text{span}\{q_1, \ldots, q_j\}$, we have to find an upper triangular matrix $\hat{R} \in \mathbb{C}^{n \times n}$ with non-zero diagonal, so that

$$A = \hat{Q}\hat{R},$$

where $\hat{Q} \in \mathbb{C}^{m \times n}$ is the matrix with columns $q_1, \ldots, q_n$. If $\{q_1, \ldots, q_n\}$ is an orthonormal set, then $A = \hat{Q}\hat{R}$ is the reduced QR factorization of A.

Like the SVD, there is a Full QR factorization. If $m \geq n$, we add $m - n$ orthonormal columns to $\hat{Q}$, making it into a unitary matrix $Q \in \mathbb{C}^{m \times m}$, while also adding $m - n$ rows of zero to $\hat{R}$, making it into an upper triangular matrix $R \in \mathbb{C}^{m \times n}$. The result $A = QR$ is the full QR factorization.



Exercise: Show that the columns $\{q_n, \ldots, q_m\}$ in Q span the complement to range(A).

MMAT 5320 Computational Mathematics - Part 1 Numerical Linear Algebra

Andrew Lam

Gram–Schmidt orthogonalization

To obtain the (reduced) QR factorization of a full ranked matrix $A \in \mathbb{C}^{m\times n}$, we have to find:

- orthonormal vectors $\{q_1, \dots, q_n\}$;
- entries $r_{ij} \in \mathbb{C}$ for $1 \le i \le m$ and $1 \le j \le n$

satisfying

$$a_1 = r_{11} q_1, \quad a_2 = r_{12} q_1 + r_{22} q_2, \quad \dots, \quad a_n = \sum_{i=1}^{n} r_{in} q_i.$$

One way is via the Gram–Schmidt orthonormalization process: Let $\{a_1, \dots, a_n\}$ be a set of LI vectors (not necessary orthogonal).

Step 1: Set $r_{11} = \|a_1\| \neq 0$ and $q_1 = \frac{a_1}{\|a_1\|} = \frac{1}{r_{11}} a_1$. Then, $\text{span}\{a_1\} = \text{span}\{q_1\}$.

Step 2: Set $v_2 = a_2 - (q_1^* a_2) q_1$ and $r_{22} = \|v_2\|$ and $q_2 = \frac{1}{r_{22}} v_2$. Then,

- $v_2 \neq 0$ and $r_{22} \neq 0$, otherwise a_2 is a linear combination of a_1!
- $q_2^* q_2 = \frac{1}{r_{22}^2} (v_2^* v_2) = 1$.
- $q_1^* q_2 = \frac{1}{r_{22}} (q_1^* v_2) = \frac{1}{r_{22}} ((q_1^* a_2) - (q_1^* a_2)) = 0$, and so $\{q_1, q_2\}$ is an orthonormal set.
- we set $r_{12} = (q_1^* a_2)$ so that $a_2 = r_{22} q_2 + r_{12} q_1$.

Gram–Schmidt orthogonalization

Step k: suppose $q_1, \dots, q_{k-1}$ have been defined and they form an orthonormal set. We now set

$$v_k = a_k - \sum_{i=1}^{k-1} (q_i^* a_k) q_i, \quad r_{kk} = \|v_k\|, \quad q_k = \frac{1}{r_{kk}} v_k,$$

and

$$r_{ik} = (q_i^* a_k) \text{ for } 1 \leq i \leq k-1.$$

Exercise: Show that $\{q_1, \dots, q_k\}$ is an orthonormal set with span$\{a_1, \dots a_k\}$ = span$\{q_1, \dots, q_k\}$.

Once all n vectors have been calculated, we have the reduced QR factorization by setting matrix $\hat{Q}$ with columns $\{q_1, \dots, q_n\}$ and upper triangular matrix $\hat{R}$ with entries r_{ij}.

In many commercial softwares, another variant of the Gram–Schmidt orthonormalization process (called modified Gram–Schmidt) is used instead of the method presented above. Since, the above method is prone to numerical instability due to rounding errors on computers.

Gram–Schmidt orthogonalization

Theorem: Every $A \in \mathbb{C}^{m \times n}$ with $m \geq n$ has a full QR factorization (and hence also a reduced QR factorization).

Proof: (1) Suppose first A has full rank, then the Gram–Schmidt algorithm gives a reduced QR factorization $A = \hat{Q}\hat{R}$.

(2) Failure can occur only if at some step j, $v_j = a_j - \sum_{i=1}^{j-1}(q_i^* a_j)q_i = 0$. But this contradicts the full rank assumption of A.

(3) Now suppose A does not have full rank, then as described above, at some step j, we have $v_j = 0$. Then, we just pick an arbitrary unit vector q_j that is orthogonal to $\{q_1, \ldots, q_{j-1}\}$ and continue the process.

(4) To get the full QR factorization, we extend the Gram–Schmidt process after step n by adding an additional $m - n$ steps, each time introducing vectors q_j that are orthonormal to $\{q_1, \ldots, q_{j-1}\}$ for $n + 1 \leq j \leq m$.

$\square$

What about the case $m < n$? Try following the Gram–Schmidt procedure.

Example

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Then, $a_1 = (1,0,1)^\top$ and $r_{11} = \|a_1\| = \sqrt{2}$, so

$$q_1 = \frac{a_1}{r_{11}} = (1/\sqrt{2}, 0, 1/\sqrt{2})^\top.$$

Next, $v_2 = a_2 - (q_1^* a_2) q_1 = (1,1,-1)^\top$ and $r_{22} = \|v_2\| = \sqrt{3}$, so

$$q_2 = \frac{v_2}{r_{22}} = (1/\sqrt{3}, 1/\sqrt{3}, -1/\sqrt{3})^\top, \quad r_{12} = q_1^* a_2 = \sqrt{2}.$$

Next, $v_3 = a_3 - \frac{1}{\sqrt{2}} q_1 + \frac{1}{\sqrt{3}} q_2 = (-1/6, 1/3, 1/6)^\top$, and $r_{33} = \|v_3\| = 1/\sqrt{6}$, so

$$q_3 = \frac{v_3}{r_{33}} = (-1/\sqrt{6}, 2/\sqrt{6}, 1/\sqrt{6})^\top, \quad r_{13} = q_1^* a_3 = 1/\sqrt{2}, \quad r_{23} = q_2^* a_3 = -1/\sqrt{3}.$$

Hence,

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{6}} \end{pmatrix} \begin{pmatrix} \sqrt{2} & \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{3} & -\frac{1}{\sqrt{3}} \\ 0 & 0 & \frac{1}{\sqrt{6}} \end{pmatrix}$$

Gram–Schmidt projections

The (classical) GS process suffers from numerical instability due to rounding errors. To overcome this, orthogonal projections are used to derive a reformulation.

Let $A \in \mathbb{C}^{m\times n}$ be of full rank and $\{a_j\}_{j=1}^n$ are the columns of A. Again the aim is to find orthonormal vectors $\{q_1, \dots, q_n\}$ such that

$$\text{span}\{a_1, \dots, a_j\} = \text{span}\{q_1, \dots, q_j\} \text{ for each } j = 1, \dots, n.$$

Suppose, we define

$$q_1 = \frac{P_1 a_1}{\|P_1 a_1\|}, \quad q_2 = \frac{P_2 a_2}{\|P_2 a_2\|}, \quad \dots, \quad q_n = \frac{P_n q_n}{\|P_n q_n\|},$$

for some orthogonal projectors $P_1, \dots, P_n$. Then, it is clear that $\|q_i\| = 1$ for $1 \le i \le n$. But what conditions do we need P_i to satisfy?

- $P_i^2 = P_i$ and $P_i^* = P_i$ [Defn. of an orthogonal projector]
- P_j projects $\mathbb{C}^m$ onto the space orthogonal to $\text{span}\{q_1, \dots, q_{j-1}\}$.

E.g., $v_2 := P_2 a_2$ will be orthogonal to $\text{span}\{q_1\}$. $v_3 = P_3 a_3$ will be orthogonal to $\text{span}\{q_1, q_2\}$, and so on... $\Rightarrow \{q_1, \dots, q_n\}$ will be an orthonormal set.

Modified Gram–Schmidt

Aim: to define

$$q_1 = \frac{P_1 a_1}{\|P_1 a_1\|}, \quad q_2 = \frac{P_2 a_2}{\|P_2 a_2\|}, \quad \dots, \quad q_n = \frac{P_n q_n}{\|P_n q_n\|},$$

with orthogonal projections P_i for $1 \leq i \leq n$ such that

- P_j projects $\mathbb{C}^m$ onto the space orthogonal to $\text{span}\{q_1, \dots, q_{j-1}\}$.

Then, each $P_j \in \mathbb{C}^{m \times m}$ must be of rank $m - (j-1)$ [(why?)]. Therefore, we can choose $P_1 = I$ the identity matrix.

For $j = 2$, we recall from Construction II that rank-one projector $A = qq^* \in \mathbb{C}^{m \times m}$ (with unit vector q) has a complement $A_\perp := I - qq^*$ of rank $m - 1$. This motivates us to choose

$$P_2 = I - q_1 q_1^* =: P_{\perp q_1}.$$

Class Exercise: Show that for two orthogonal unit vectors q_1 and q_2, the matrix X defined as

$$X = P_{\perp q_2} P_{\perp q_1} = (I - q_2 q_2^*)(I - q_1 q_1^*)$$

is an orthogonal projector which projects $\mathbb{C}^m$ onto the space orthogonal to $\text{span}\{q_1, q_2\}$. What is the rank of X?

Modified Gram–Schmidt

Hence, for each $j \in \{2, \dots, n\}$ we define

$$P_j = P_{\perp q_j} P_{\perp q_{j-1}} \dots P_{\perp q_1}$$

with $P_1 = I$. Then, $\{P_1, \dots, P_n\}$ satisfies the required properties.

For example, given $\{q_1, \dots, q_{j-1}\}$, in order to obtain q_j, we perform the calculations in the following order:

$$\begin{aligned}
v_j^{(1)} &= a_j, \\
v_j^{(2)} &= P_{\perp q_1} v_j^{(1)} = v_j^{(1)} - (q_1 q_1^*) v_j^{(1)}, \\
v_j^{(3)} &= P_{\perp q_2} v_j^{(2)} = v_j^{(2)} - (q_2 q_2^*) v_j^{(2)}, \\
&\ \vdots \\
v_j := v_j^{(j)} &= P_{\perp q_{j-1}} v_j^{(j-1)} = v_j^{(j-1)} - (q_{j-1} q_{j-1}^*) v_j^{(j-1)}, \\
q_j &: = v_j / \|v_j\|.
\end{aligned}$$

Then, one defines

$$r_{jj} = \|v_j\|, \quad r_{ij} = q_i^* a_j \text{ for } 1 \le i \le j-1.$$

Another interpretation of modified Gram–Schmidt

In practice, it is common to initialise $v_i = a_i$ and later overwrite them after computations to save storage. Each step of the modified Gram–Schmidt algorithm can be interpreted as a right-multiplication by a square upper-triangular matrix.

E.g., at the first step, we multiply first column a_1 by $1/r_{11}$ where $r_{11} = \|a_1\|$, and then subtract r_{1j} times the result from each of the remaining columns a_j. This is equivalent to right multiplication by a matrix R_1:

$$\left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right] \begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \frac{-r_{13}}{r_{11}} & \cdots \\ & 1 & & \\ & & 1 & \\ & & & \ddots \end{bmatrix} = \left[\begin{array}{c|c|c|c} q_1 & v_2^{(2)} & \cdots & v_n^{(2)} \end{array}\right]$$

Here on the left we have set $v_i = a_i$.

The next steps are similar.

At the ith step, we subtract r_{ij}/r_{ii} times column i of the current matrix from columns $j > i$, and replace column i by $1/r_{ii}$ times itself. This corresponds to multiplication with upper-triangular matrices R_i of the form

$$R_2 = \begin{bmatrix} 1 & & & \\ & \frac{1}{r_{22}} & \frac{-r_{23}}{r_{22}} & \cdots \\ & & 1 & \\ & & & \ddots \end{bmatrix}, \quad R_3 = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \frac{1}{r_{33}} & \cdots \\ & & & \ddots \end{bmatrix}, \cdots .$$

At the end of the iteration we have

$$A \underbrace{R_1 R_2 \cdots R_n}_{=:\hat{R}^{-1}} = \hat{Q}$$

leading to the reduced QR factorization of A.

Householder triangularization

While (modified) Gram–Schmidt is a feasible method to compute the QR factorization. We introduce another method which is numerically more stable.

In previous slide, the modified Gram–Schmidt algorithm can be seen as applying a succession of upper triangular matrices R_k on the right of A so that

$$AR_1 \dots R_n = \hat{Q} \in \mathbb{C}^{m \times n}$$

has orthonormal columns. Setting $\hat{R} = R_n^{-1} \cdots R_1^{-1}$ we get the reduced QR factorization $A = \hat{Q}\hat{R}$. Hence, GS is the method of triangular orthogonalization.

Householder's method instead applies a succession of unitary matrices Q_k on the left of A, so that

$$Q_n \cdots Q_1 A = R$$

is upper triangular. The matrix

$$Q := Q_1^* Q_2^* \cdots Q_n^*$$

is unitary and we get the full QR factorization $A = QR$. Hence, Householder's method is orthogonal triangularization.

Idea of Householder's method

- Multiplying A with Q_1 should reduces all entries below (1,1) entry in the first column to zero.
- Then, multiplying with Q_2 should reduce all entries below the (2,2) entry in the second column of $Q_1 A$ to zero,
- and so on ...

For example, if A is a 5×3 matrix.

$$\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1 A}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{Q_2} \underset{Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} \end{bmatrix}} \xrightarrow{Q_3} \underset{Q_3 Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & & \boldsymbol{\times} \\ & & \mathbf{0} \\ & & \mathbf{0} \end{bmatrix}}$$

In general, Q_k only operates on rows $k, \ldots, m$. After n steps (assuming here $m \geq n$), all entries below the main diagonal would have been eliminated, and $Q_n \cdots Q_1 A = R$ is then an upper triangular matrix.

Finding the unitary matrices

To find these unitary matrices, we choose them to be of the form

$$Q_k = \begin{pmatrix} I_{(k-1)\times(k-1)} & 0 \\ 0 & F \end{pmatrix}$$

where $F \in \mathbb{C}^{(m-k+1)\times(m-k+1)}$ is a unitary matrix.

- Note that multiplication by Q_k leaves the first $k-1$ rows unchanged.
- We want the multiplication by F to change the kth column as intended in the Householder method., i.e., it create zeros below the kth diagonal entry. This is done by so-called Householder reflectors.

Idea: let $x \in \mathbb{C}^{m-k+1}$ be the (sub)vector containing the $(k,k),\dots,(k,m)$ entries from the kth column. The action of F should look like

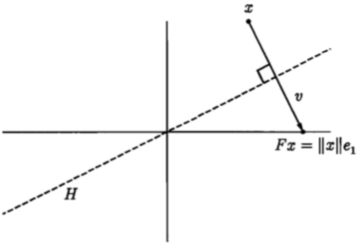
$$x = \begin{bmatrix} \times \\ \times \\ \times \\ \vdots \\ \times \end{bmatrix} \quad \xrightarrow{F} \quad Fx = \begin{bmatrix} \|x\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1.$$

where $e_1 = (1,0,\dots,0)^\top \in \mathbb{C}^{m-k+1}$.

Householder reflectors I

Geometric picture for $x \in \mathbb{R}^2$, i.e., $m - k + 1 = 2$: We want to transform

$$x = \begin{pmatrix} \times \\ \times \end{pmatrix} \quad \xrightarrow{F} \quad Fx = \begin{pmatrix} \|x\| \\ 0 \end{pmatrix} = \|x\| e_1.$$

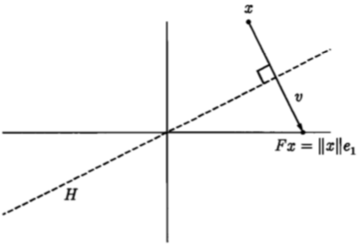


The vector $v = \|x\| e_1 - x$ generates a (hyper)plane H (which is orthogonal to v), so that when we reflect point x across H, we land at Fx.

Then, if v is a unit vector,

$$Fx = x - 2(v^* x)v = (I - 2vv^*)x, \quad \text{i.e.,} \quad F = I - 2vv^*.$$

Householder reflector II



More generally, we set

$$F = I - 2\frac{vv^*}{\|v\|^2} = I - \frac{2vv^*}{v^*v} \text{ for } v = \|x\|e_1 - x.$$

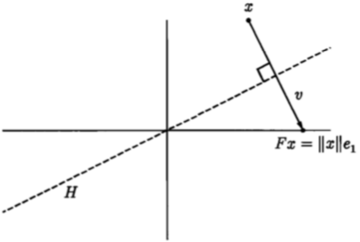
Recall the orthogonal projector P defined by

$$P = I - \frac{vv^*}{\|v\|^2}$$

which projects a vector $w \in \mathbb{C}^2$ (in this picture) to the plane orthogonal to v. In comparison, we need to go twice in the direction of v to get to the point Fx.

Householder reflectors III

In implementations, computations can become unstable during subtraction. For example, if the angle between H and the e_1 axis is small, then the magnitude of $\|x\|e_1$ and x are very close, and computing the subtraction $v = \|x\|e_1 - x$ may lead to loss of significance/unwanted cancellation.

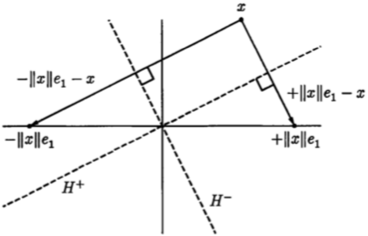


To remedy this, the general rule is to reflect the vector x to $z\|x\|e_1$, where z is any scalar with $|z| = 1$.

- In the real case, $|z| = 1$ implies $z = \pm 1$;
- In the complex case, there is a circle of possibilities for z

Householder reflectors IV

In the real case, there are two reflections: across H^+ and across H^-:



Reflection across $H^{\pm}$ gives $F^{\pm} = I - \frac{2v_{\pm}v_{\pm}^*}{\|v_{\pm}\|_2^2}$ where $v_{\pm} = \pm\|x\|e_1 - x$.

To avoid numerical instability, it is recommended to choose to reflect x to the vector that is not too close to x itself, i.e., choose v so that $\|v\|$ is large. For example, we can choose (where x_1 is the 1st component of x)

$$z = \begin{cases} -\text{sign}(x_1) & \text{if } x_1 \neq 0, \\ 1 & \text{if } x_1 = 0, \end{cases} \quad \text{sign}(y) = \begin{cases} 1 & \text{Re}(y) > 0, \\ -1 & \text{Re}(y) < 0, \\ \text{sign}(\text{Im}(y)) & \text{Re}(y) = 0, \end{cases}$$

and set $v = -\text{sign}(x_1)\|x\|e_1 - x$.

Exercise: It is customary to use $v = \text{sign}(x_1)\|x\|e_1 + x$. Show that this gives the same Householder reflector F.

Householder QR factorization

The Householder algorithm computes the triangular matrix R of the full QR factorization of a matrix $A \in \mathbb{C}^{m\times n}$ (for $m \geq n$):

Step 1: Set $x = a_1$ as the first column of A, construct vector $v_1 = \text{sign}(x_1)\|x\|e_1 + x$ where $e_1 = (1, 0, \dots, 0)^\top \in \mathbb{C}^m$, and Householder reflector $F_1 = I - \frac{2v_1 v_1^*}{\|v_1\|^2} \in \mathbb{C}^{m\times m}$. Then,

$$Q_1 = F_1.$$

Step 2: Set $x = \hat{a}_2 = (\hat{a}_{22}, \dots, \hat{a}_{2n})^\top \in \mathbb{C}^{m-1}$ as the second column of $Q_1 A$ without the first entry. Construct vector $v_2 = \text{sign}(x_1)\|x\|e_1 + x$ where $e_1 = (1, 0, \dots, 0)^\top \in \mathbb{C}^{m-1}$, and Householder reflector $F_2 = I - \frac{2v_2^* v_2^*}{\|v_2\|^2} \in \mathbb{C}^{(m-1)\times(m-1)}$. Then,

$$Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & F_2 \end{pmatrix}.$$

Step k: Set $x \in \mathbb{C}^{m-k+1}$ as the kth column of $Q_{k-1}\cdots Q_1 A$ without the first $k-1$ entries. Construct vector $v_k \in \mathbb{C}^{m-k+1}$ and Householder reflector $F_k \in \mathbb{C}^{(m-k+1)\times(m-k+1)}$. Then,

$$Q_k = \begin{pmatrix} I_{k-1\times k-1} & 0 \\ 0 & F_k \end{pmatrix}.$$

Then, $Q = Q_1 Q_2 \cdots Q_n$ and $R = Q^* A$.

Householder algorithm (implementation)

In practice, there is no need to compute matrices $Q_1, \dots, Q_n$ as described above. A pseduocode for the Householder QR factorization would be the following:

Notation: If A is a matrix, then we set $A_{i_1:i_2, j_1:j_2}$ to be the submatrix of size $(i_2 - i_1 + 1) \times (j_2 - j_1 + 1)$ with upper-left corner a_{i_1, j_1} and lower-right corner a_{i_2, j_2}. In the case the submatrix reduces to a subvector of a row or column we write $A_{i, j_1:j_2}$ or $A_{i_1:i_2, j}$.

$$
\begin{aligned}
&\textbf{for } k = 1 \textbf{ to } n \\
&\qquad x = A_{k:m,k} \\
&\qquad v_k = \operatorname{sign}(x_1)\|x\|_2 e_1 + x \\
&\qquad v_k = v_k / \|v_k\|_2 \\
&\qquad A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^* A_{k:m,k:n})
\end{aligned}
$$

This updates the matrix A into the upper triangular matrix R while storing the n reflection vectors $v_1, \dots, v_n$.

Class exercise

1. Show that for a unit vector v, the Householder matrix $P = I - 2vv^*$ for a unit vector v is hermitian, unitary, and has eigenvalues ± 1.
2. Compute the singular values of P.
3. Show that P has determinant equal to -1.
4. Compute the full/reduced QR factorization of

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 7 \\ 4 & 2 & 3 \\ 4 & 2 & 2 \end{pmatrix}$$

   using Householder or Gram–Schmidt.
5. Given knowledge of the reflection vectors $v_1, \dots, v_n$, write down a code to compute the product $Q^* b$ for an arbitrary $b \in \mathbb{R}^m$ without explicitly constructing the matrix Q.

Matrix-vector problem

Problem: Given a matrix $A \in \mathbb{C}^{m\times n}$ and vector $b \in \mathbb{C}^m$, find a solution $x \in \mathbb{C}^n$ to the equation $Ax = b$.

- If $m = n$, then there is a (unique) solution if A is invertible.
- If $m > n$, $\#$ equ. $> \#$ unknowns, i.e., the system is overdetermined, and typically there is no solutions.
- If $m < n$, $\#$ equ. $< \#$ unknowns, i.e., the system if underdetermined, and typically there is an infinite number of solutions.

Example:

$$2x = 6, \quad 3x = 6 \quad \Leftrightarrow \quad Ax = b \text{ with } A = \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \quad b = \begin{pmatrix} 6 \\ 6 \end{pmatrix}$$

has no solution. While

$$2x_1 + 3x_2 = 5 \quad \Leftrightarrow \quad Ax = b \text{ with } A = \begin{pmatrix} 2 & 3 \end{pmatrix}, \quad b = \begin{pmatrix} 5 \end{pmatrix}$$

has infinitely many solutions of the form $(t, \frac{1}{3}(5 - 2t))$ for any $t \in \mathbb{R}$.

Overdetermined case

The undetermined case can be partially solved by selecting, amongst all possible solutions, one that has the smallest norm, e.g., the 2-norm.

For the overdetermined case, the residual

$$r = b - Ax \in \mathbb{C}^m$$

will never be zero, but an acceptable solution to $Ax = b$ would be a vector x_* whose residual is the smallest w.r.t to some norm.

Choosing the 2-norm leads to the general least squares problem: Given $A \in \mathbb{C}^{m\times n}$ with $m \geq n$, and $b \in \mathbb{C}^m$, find $x_* \in \mathbb{C}^n$ such that

$$\|b - Ax_*\|_2 \leq \|b - Ay\|_2 \text{ for all } y \in \mathbb{C}^n.$$

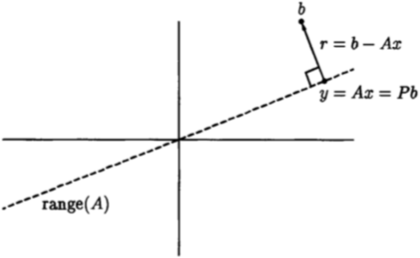
Equivalently,

$$\|b - Ax_*\|_2 \leq \|b - z\|_2 \text{ for all } z \in \text{range}(A).$$

Solving the least squares problem

Goal: Find $Ax \in \text{range}(A)$ closest to b.

Geometrically: the answer y is equal Pb, where P is the projection onto range(A).



So, we need to find x such that $Ax = y = Pb$ in order to solve the least squares problem.

Solving the least squares problem II

Theorem: Let $A \in \mathbb{C}^{m \times n}$ with $m \geq n$ and $b \in \mathbb{C}^m$ be given. A vector $x \in \mathbb{C}^n$ solves the least squares problem if and only if it solves the normal equation

$$A^* A x = A^* b.$$

Proof: (1) If x is a solution, then residual $r = b - Ax$ is orthogonal to range(A). This means

$$A^* r = 0 \quad \Rightarrow \quad A^*(b - Ax) = 0.$$

(2) If x solves the normal equation, then $A^*(b - Ax) = 0$, and this means its residual $r = b - Ax$ is orthogonal to range(A). Let's define $P \in \mathbb{C}^{m \times m}$ as the orthogonal projector onto range(A). Then, (recall the geometric picture)

$$Pb = Ax.$$

(3) Let $z \in$ range(A) be arbitrary and set $y = Pb$. Then, $b - y = r$ is orthogonal to $z - y \in$ range(A), and so by Pythagorean theorem

$$\|b - z\|_2^2 = \|b - y\|_2^2 + \|y - z\|_2^2 \geq \|b - y\|_2^2,$$

which means $y = Pb = Ax$ solves the least squares problem. $\square$

Supplementary

Pythagorean theorem: If x and y are orthogonal, then

$$\|x+y\|_2^2 = \|x\|_2^2 + \|y\|_2^2.$$

Exercise: Prove this.

Exercise: Show that if A has full rank if and only if A^*A is invertible. Consequently, deduce that the solution to the normal equation is unique if and only if A has full rank.

Pseudoinverse

Let $A \in \mathbb{C}^{m \times n}$ $(m \geq n)$ be of full rank. Then, an acceptable solution x_* to an overdetermined system $Ax = b$ is the solution to the least squares problem:

$$\|b - Ax_*\|_2 \leq \|b - Ay\|_2 \text{ for all } y \in \mathbb{C}^m.$$

The previous theorem shows x^* can be computed from the normal equation

$$x_* = (A^*A)^{-1}A^*b.$$

This motivates defining $(A^*A)^{-1}A^*$ as the pseudoinverse A^+.

- Note that $A^+ = (A^*A)^{-1}A^* \in \mathbb{C}^{n \times m}$, as it maps $b \in \mathbb{C}^m$ to $x_* \in \mathbb{C}^n$, i.e., if $m > n$, then A^+ has more columns than rows.
- If $A \in \mathbb{C}^{m \times m}$ is invertible, then

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

  Pseudoinverse coincides with the usual inverse.

QR method for least squares

The classical way to solve the least squares problem is to solve the normal equation $A^*Ax = A^*b$.

- Good when A is full rank.
- But calculations can become unstable with rounding (small rounding errors can grow large).

The modern classical method is to use reduced QR factorization (Gram–Schmidt/Householder). Construct $A = \hat{Q}\hat{R}$, and the orthogonal projector $P = \hat{Q}\hat{Q}^*$. Then, $y = Pb = \hat{Q}\hat{Q}^*b$, and

$$Ax = Pb \quad \Rightarrow \quad \hat{Q}\hat{R}x = \hat{Q}\hat{Q}^*b \quad \Rightarrow \quad x = \hat{R}^{-1}(\hat{Q}^*b).$$

As $\hat{R}$ is upper-triangular, $\hat{R}^{-1}$ is easy to compute! This also gives another formula for the pseudoinverse[2]

$$A^+ = \hat{R}^{-1}\hat{Q}^*.$$

- Nowadays the standard method. Good when A is full rank.
- Less ideal if A is rank-deficient.

[2] typo in previous version, don't forget Q^*

SVD method for least squares

For rank-deficient matrices, we can compute the reduced SVD $A = \hat{U}\hat{\Sigma}V^*$. The orthogonal projection P is now $P = \hat{U}\hat{U}^*$ and

$$y = Pb = \hat{U}\hat{U}^* b.$$

Then,

$$Ax = Pb \quad \Rightarrow \quad \hat{U}\hat{\Sigma}V^* = \hat{U}\hat{U}^* b \quad \Rightarrow \quad x = V\hat{\Sigma}^{-1}\hat{U}^* b.$$

This gives another formula for the pseudoinverse

$$A^+ = V\hat{\Sigma}^{-1}\hat{U}^*.$$

Comparison with QR method:

- QR factorization reduces the least squares problem to solving a triangular system of equations (solve $\hat{R}x = \hat{Q}^* b$).
- SVD reduces the problem to a diagonal system of equations (solve $\hat{\Sigma}w = \hat{U}^* b$ and then set $x = Vw$).

Class exercise

1. Looking back at the slide Construction III explain why the orthogonal projection P to range(A) for the reduced QR factorization is $P = \hat{Q}\hat{Q}^*$, and why for the SVD factorization it is $P = \hat{U}\hat{U}^*$.

2. Solve the overdetermined system $Ax = b$ with

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 7 \\ 4 & 2 & 3 \\ 4 & 2 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ 2 \\ 1 \\ 1 \\ 5 \end{pmatrix}$$

   with (i) the normal equation, (ii) the QR method, (iii) the SVD method. What can you say about the corresponding solutions from each of these methods?

§5 - Eigenvalue problems

Review

Let $A \in \mathbb{C}^{m \times m}$ be a square matrix. A nonzero vector $x \in \mathbb{C}^m$ is an eigenvector of A corresponding to an eigenvalue $\lambda \in \mathbb{C}$ if

$$Ax = \lambda x.$$

The set of all eigenvalues of A is called the spectrum, denoted by $\Lambda(A)$.

To find eigenvalues, the standard way is to compute the charateristic polynomial

$$p_A(x) = \det(zI - A)$$

which is a polynomial of degree m, and search for its roots. Namely λ is an eigenvalue of A if and only if $p_A(\lambda) = 0$.

Each λ eigenvalue has two notion of mulitplicity:

- algebraic multiplicity - the number of times λ appears as a repeated root of p_A.
- geometric multiplicity - the number of LI eigenvectors corresponding to λ, aka the dimension of the eigenspace of λ.

An eigenvalue is called simple if its algebraic multiplicity is 1.

Review II

Example:

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & -1 & 1 \\ 2 & 0 & -1 \end{pmatrix}$$

has a characteristic polynomial $p_A(z) = -(z-3)(z+1)^2$.

So the eigenvalues of A are $\lambda_1 = 3$ and $\lambda_2 = \lambda_3 = -1$. The eigenvector for $\lambda_1 = 3$ is obtained by solving $(A - 3I)v_1 = 0$ which gives $v_1 = (1, 1/2, 1)^\top$. For $\lambda_2 = \lambda_3 = -1$, we see that

$$0 = (A+I)w = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 0 & 1 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} 2w_1 + 2w_2 + w_3 \\ w_1 + w_3 \\ 2w_1 + 2w_3 \end{pmatrix}.$$

We can choose $v_2 = (1, -1/2, -1)^\top$ as an eigenvector. Is there more?

Unfortunately, there is no other choice of w, therefore

- $\lambda = 3$ has algebraic and geometric multiplicity 1,
- $\lambda = -1$ has algebraic multiplicity 2 and geometric multiplicity 1.

Eigenvalue decomposition

Let $A \in \mathbb{C}^{m\times m}$ be a square matrix. Its eigenvalue decomposition is the factorization

$$A = X\Lambda X^{-1},$$

where $X \in \mathbb{C}^{m\times m}$ is invertible and $\Lambda \in \mathbb{C}^{m\times m}$ is diagonal.

This factorization is not guaranteed to exist for general square matrices!

Rewriting as $AX = \Lambda X$ yields the graphically picture

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix} \left[\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_m \\ & & & \end{array}\right] = \left[\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_m \\ & & & \end{array}\right] \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

or equivalently

$$Ax_j = \lambda_j x_j.$$

Similarity transformations

If $X \in \mathbb{C}^{m \times m}$ is invertible, the transformation

$$A \mapsto X^{-1}AX$$

is called a similarity transformation.

Related – similar matrices: $A \in \mathbb{C}^{m \times m}$ and $B \in \mathbb{C}^{m \times m}$ are similar if there exists an invertible $X \in \mathbb{C}^{m \times m}$ such that $B = X^{-1}AX$.

Also related – equivalent matrices: $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{m \times n}$ are equivalent if there exist invertible $P \in \mathbb{C}^{n \times n}$ and invertible $Q \in \mathbb{C}^{m \times m}$ such that $B = Q^{-1}AP$.

Note:

- Similarity only defined for square matrices!
- Similar $\Rightarrow$ Equivalent, but not the other way round.
- A and B are equivalent means there are bases $\mathcal{B}_1$ of $\mathbb{C}^n$ and $\mathcal{B}_2$ of $\mathbb{C}^m$ such that B is the matrix A w.r.t these new bases.
- A and B are similar means there is a basis $\mathcal{B}$ of $\mathbb{C}^m$ $(m = n)$ such that B is the matrix A w.r.t the basis $\mathcal{B}$.

Properties of similarity transformation

Theorem: If X is invertible, then A and $X^{-1}AX$ have the same characteristic polynomial. Hence, they have the same eigenvalues, and the same algebraic/geometric multiplicity.

Proof: (1) A straightforward calculation

$$\begin{aligned} p_{X^{-1}AX}(z) &= \det(zI - X^{-1}AX) = \det(X^{-1}(zI - A)X) \\ &= \det(X^{-1})\det(zI - A)\det(X) = p_A(z). \end{aligned}$$

Therefore, A and $X^{-1}AX$ have the same eigenvalues and algebraic multiplicity.

(2) Notice y is an eigenvector of A if and only if $X^{-1}y$ is an eigenvector of $X^{-1}AX$, since

$$Ay = \lambda y \quad \Leftrightarrow \quad X^{-1}AX(X^{-1}y) = \lambda(X^{-1}y).$$

Also $\{y_i\}_i$ are LI if and only if $\{X^{-1}y_i\}_i$ are LI. Therefore, the geometric multiplicity for each eigenvalue of A and $X^{-1}AX$ also agree. $\square$

Algebraic multiplicity $\geq$ Geometric multiplicity

Theorem: The algebraic multiplicity of an eigenvalue is greater than/equal to its geometric multiplicity.

Proof: (1) Suppose the geometric multiplicity of λ is n. Then, there are n LI eigenvectors $\{v_1, \dots, v_n\}$ corresponding to λ.

(2) To the set $\{v_1, \dots, v_n\}$ we add orthogonal vectors $\{v_{n+1}, \dots, v_m\}$ such that the resulting matrix $V \in \mathbb{C}^{m \times m}$ with ith column v_i is invertible. Then, defining $B = V^{-1}AV$ we see

$$B = \begin{pmatrix} \lambda I & C \\ 0 & D \end{pmatrix}$$

for some $C \in \mathbb{C}^{n \times (m-n)}$ and $D \in \mathbb{C}^{(m-n)\times(m-n)}$

(3) By properties of the determinant, and as A and B are similar,

$$p_A(z) = \det(zI - A) = \det(zI - B) = (z - \lambda)^n \det(zI - D).$$

As we cannot rule out if λ is also a root to $p_D(z)$, the algebraic multiplicity of λ is at least n. $\square$

Defective matrices

An eigenvalue of a matrix $A \in \mathbb{C}^{m \times m}$ is called defective if its algebraic multiplicity $>$ its geometric multiplicity.

A matrix is called defective if it has at least one defective eigenvalue. Otherwise it is non-defective.

Theorem: A diagonal matrix is non-defective.

Proof: Let λ be an eigenvalue of the diagonal matrix A. Then,

$$\text{algebraic multiplicity} = \text{no. of times it appears on the diagonal.}$$

An eigenvector for entry $\lambda_i = a_{ii}$ is the unit vector e_i, and $\{e_1, \dots, e_m\}$ are LI. Therefore

$$\text{geometric multiplicity} = \text{no. of times it appears on the diagonal.}$$

$\square$

Diagonalizability

Theorem: $A \in \mathbb{C}^{m\times m}$ is non-defective if and only if it has an eigenvalue decomposition $A = X\Lambda X^{-1}$ with X invertible and Λ diagonal.

Proof: ($\Rightarrow$) A non-defective implies A has m LI eigenvectors. Let X be the matrix whose columns are these eigenvectors. Then, X is invertible and $AX = X\Lambda$, whre Λ is the diagonal matrix with entries equal to the eigenvalues of A.

($\Leftarrow$) Since Λ is diagonal it is non-defective, and as the eigenvalue decomposition means A is similar to Λ, and similarity transformation preserves algebraic and geometric multiplicities of eigenvalues, we must have A is also non-defective. $\square$

This result motivates the definition: $A \in \mathbb{C}^{m\times m}$ is diagonalizable $\Leftrightarrow$ it has an eigenvalue decomposition $A = X\Lambda X^{-1}$ $\Leftrightarrow$ it is non-defective.

Unitary diagonalizability

If $A = X\Lambda X^{-1}$ and the columns of X is orthonormal, then X is a unitary matrix, i.e., $X^* = X^{-1}$ (why?).

In this case, we say $A \in \mathbb{C}^{m\times m}$ is unitary diagonalizable if $A = Q\Lambda Q^*$ for some unitary matrix Q and diagonal matrix Λ.

Note:

- The "eigenvalue decomposition" $A = Q\Lambda Q^*$ can also be seen as a SVD for the square matrix A.
- If A is hermitian, i.e., $A = A^*$, then it is unitary diagonalizable and all entries in Λ are real. (To be proven later...)

Schur factorization

The Schur factorization of a square matrix $A \in \mathbb{C}^{m \times m}$ is of the form

$$A = QTQ^*$$

where Q is unitary and T is upper-triangular.

Note: A and T are similar $\Rightarrow$ eigenvalues of A appear on the diagonal of T.

Theorem: Every square matrix has a Schur factorization.

Proof: Induction on m. Case $m = 1$ is trivial. So suppose $m \geq 2$.

(1) Let $x \in \mathbb{C}^m$ be any eigenvector of A with eigenvalue λ. Normalize x and set it to be the first column of a unitary matrix $U \in \mathbb{C}^{m \times m}$. From the slide Algebriac multiplicity $\geq$ Geometric multiplicity, we have

$$U^*AU = \begin{pmatrix} \lambda & b \\ 0 & C \end{pmatrix} \text{ with } b^\top \in \mathbb{C}^{m-1} \text{ and } C \in \mathbb{C}^{m-1 \times m-1}.$$

(2) By induction hypothesis, since $C \in \mathbb{C}^{m-1 \times m-1}$, it has a Schur factorization, $C = VTV^*$ with upper-triangular matrix $T \in \mathbb{C}^{m-1 \times m-1}$ and unitary V.

(3) Then, the matrix Q defined below is unitary

$$Q = U \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix} \quad \Rightarrow \quad Q^*AQ = \begin{pmatrix} 1 & 0 \\ 0 & V^* \end{pmatrix} \begin{pmatrix} \lambda & b \\ 0 & C \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & V \end{pmatrix} = \begin{pmatrix} \lambda & bV \\ 0 & T \end{pmatrix}$$

$\square$

Consequences of Schur factorization

Theorem: Every hermitian matrix is unitary diagonalizable, and so all its eigenvalues are real.

Proof: Let A be hermitian, i.e., $A = A^*$. Then, by Schur factorization,

$$A = QTQ^* = (QTQ^*)^* = A^* \quad \Rightarrow \quad QTQ^* = QT^*Q^*.$$

Hence, $T = T^*$ which implies T must be diagonal. Furthermore, on the diagonal, it holds that $T_{ii} = \overline{T_{ii}}$, which implies the diagonal entries of T are real numbers. $\square$

We say that a square matrix A is normal if $AA^* = A^*A$.

Theorem: A square matrix is unitary diagonalizable if and only if it is normal.

Proof: ($\Rightarrow$) If $A = Q\Lambda Q^*$, then $A^* = Q\Lambda^* Q^*$ and

$$AA^* = Q\Lambda\Lambda^* Q^* = Q\Lambda^*\Lambda Q^* = A^*A.$$

($\Leftarrow$) Let $A = UTU^*$ be its Schur factorization. As A is normal

$$UTT^*U^* = AA^* = A^*A = UT^*TU.$$

This implies $TT^* = T^*T$, i.e., the upper triangular matrix T is also normal.

Consequences of Schur factorization II

If

$$T = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1m} \\ & t_{22} & \cdots & t_{2m} \\ & & \ddots & \vdots \\ & & & t_{mm} \end{pmatrix}$$

then, the (1,1) entry of T^*T and TT^* are respectively

$$|t_{11}|^2 \quad \text{and} \quad \sum_{i=1}^{m} |t_{1i}|^2,$$

and $TT^* = T^*T$ implies $|t_{1i}| = 0$ for $2 \leq i \leq m$, i.e., the first row of T is zero except for the (1,1)-entry.

Next, the (2,2)-entry of T^*T and TT^* are respectively

$$|t_{12}|^2 + |t_{22}|^2 = |t_{22}|^2 \quad \text{and} \quad \sum_{i=2}^{m} |t_{2i}|^2.$$

Again, $TT^* = T^*T$ implies $|t_{2i}| = 0$ for $3 \leq i \leq m$.

Continue this way, all upper off diagonal entries of T are zero, and so T is a diagonal matrix. $\square$

Class exercise

1. Show that the matrix $I - vv^*$ is unitary if and only if $\|v\|_2^2 = 2$ or $v = 0$.
2. Show that these two matrices are not similar
$$A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$
3. Show that the following matrix is singular, but is diagonalizable
$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
4. Show that the following matrix is nonsingular, but is not diagonalizable
$$A = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$
5. Find the Schur factorization of the matrices in Q3 and Q4.

§6 - Eigenvalue algorithms

Eigenvalue revealing methods

Methods for finding eigenvalues of $A \in \mathbb{C}^{m \times m}$:

1. Solving the characteristic polynomial p_A.
   - unfeasible for large size matrices.
   - (rounding) errors in coefficients $\Rightarrow$ inaccurate calculations.

2. Iterative process to find the largest eigenvalue, e.g., the Power iteration.

   Idea: the sequence

   $$x, \quad Ax, \quad A^2x, \quad A^3x, \quad \cdots$$

   will converge (under certain conditions) to an eigenvector associated with the largest eigenvalue of A in magnitude.
   - A similar method (inverse power iteration) computes the smallest eigenvalue of A in magnitude.
   - Depends strongly on well-separated eigenvalues, e.g., $|\lambda_2|/|\lambda_1| \ll 1$.
   - Convergence can be rather slow otherwise.

3. Factorise A so that its eigenvalues appear in one of the factors:
   - Diagonalization for non-defective $A = X\Lambda X^{-1}$. Eigenvalues listed in diagonal matrix Λ.
   - Unitary diagonalizability for non-defective $A = Q\Lambda Q^*$.
   - Schur factorization for any $A = QTQ^*$. Eigenvalues appear on diagonal of the upper triangular matrix T.

Difficulty with characteristic polynomial

A deep result in Galois theory:

Theorem: For any $m \geq 5$, there is a polynomial $p(z)$ of degree m with rational coefficients that has a real root $p(r) = 0$ with the property that r cannot be written using any expression involving rational numbers, addition, subtraction, multiplication, division, and kth roots.

Meaning? There is no analogue of the quadratic formula for polynomials of degree ≥ 5.

Which means? There is no compute program that would product the exact roots of an arbitrary polynomial in a finite number of steps.

Hence, any eigenvalue solver must be iterative, i.e., generate a sequence of numbers that converges (rapidly) towards eigenvalues.

Power iteration

Designed to compute the dominant eigenvalue of a matrix $A \in \mathbb{C}^{m \times m}$ and an associated eigenvector.

Assumptions:

- There is a single eigenvalue of maximum modulus. I.e., we can label the eigenvalues in terms of their magnitude:

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_m|.$$

- There is a set of m LI eigenvectors. I.e., there is a basis $\{u_1, \ldots, u_m\}$ of $\mathbb{C}^m$ such that

$$Au_j = \lambda_j u_j \text{ for } 1 \leq j \leq m.$$

Procedure: Pick an arbitrary initial vector $x_0 \in \mathbb{C}^m$. Generate sequences

- $z_k = Ax_k$,
- $x_{k+1} = \frac{z_k}{\|z_k\|_2}$,
- $r_{k+1} = x_{k+1}^* A x_{k+1}$.

Theorem: If the initial vector x_0 has an expansion of the form $x_0 = a_1 u_1 + \cdots + a_m u_m$ with $a_1 \neq 0$, then as $k \to \infty$, x_{k+1} aligns along with direction of u_1, and

$$r_k \to \lambda_1 \text{ as } k \to \infty.$$

Power iteration II

Proof: (1) From the definition, we see that

$$x_k = \frac{A^k x_0}{\|A^k x_0\|_2} \text{ for } k \geq 1.$$

Then, by the expansion of x_0,

$$A^k x_0 = a_1 \lambda_1^k \left(u_1 + \sum_{i=2}^{m} \frac{a_i}{a_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k u_i \right) =: a_1 \lambda_1^k (u_1 + \varepsilon_k) \text{ for } k \geq 1.$$

(2) Since $|\lambda_1| > |\lambda_j|$ for $j \geq 2$, $\frac{\lambda_j^k}{\lambda_1^k}$ converges to zero, and so the vector $\varepsilon_k \to 0$ as $k \to \infty$. Therefore,

$$x_k = \frac{A^k x_0}{\|A^k x_0\|_2} = \frac{a_1 \lambda_1^k (u_1 + \varepsilon_k)}{\|a_1 \lambda_1^k (u_1 + \varepsilon_k)\|_2} = \operatorname{sign}(a_1 \lambda_1^k) \frac{u_1 + \varepsilon_k}{\|u_1 + \varepsilon_k\|_2},$$

i.e., x_k aligns more and more with the direction of u_1 as $k \to \infty$.

(3) Next,

$$r_k = x_k^* A x_k = \frac{(u_1 + \varepsilon_k)^* (\lambda_1 u_1 + A \varepsilon_k)}{\|u_1 + \varepsilon_k\|_2^2} \to \lambda_1.$$

$\square$

Power iteration III

Theorem: Let $A \in \mathbb{C}^{m\times m}$ be diagonalizable with eigenvalues

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_m|$$

and normalized eigenvectors $\{u_1, \dots, u_m\}$. Let $x_0 = a_1 u_1 + \cdots + a_m u_m$ be any vector with $a_1 \neq 0$. Then, there is a constant $c > 0$ such that

$$\|y_k - u_1\|_2 \leq c \left| \frac{\lambda_2}{\lambda_1} \right|^k \quad \text{for } y_k = \frac{x_k \|A^k x_0\|_2}{a_1 \lambda_1^k}.$$

Meaning: If $|\lambda_2|$ is close to $|\lambda_1|$, convergence of sequence y_k to eigenvector u_1 is slow.

Proof: Short computation

$$\begin{aligned}\|y_k - u_1\|_2 &= \left\| \sum_{i=2}^m \frac{a_i \lambda_i^k}{a_1 \lambda_1^k} u_i \right\|_2 \\ &\leq \Big(\sum_{i=2}^m \left[\frac{a_i}{a_1}\right]^2 \left[\frac{\lambda_i}{\lambda_1}\right]^{2k} \Big)^{1/2} \leq \left|\frac{\lambda_2}{\lambda_1}\right|^k \Big(\sum_{i=2}^m \left[\frac{a_i}{a_1}\right]^2 \Big)^{1/2} =: c \left|\frac{\lambda_2}{\lambda_1}\right|^k\end{aligned}$$

$\square$

Inverse power iteration

Theorem: If λ is an eigenvalue of A and if A is invertible, then $\frac{1}{\lambda}$ is an eigenvalue of A^{-1}.

Exercise: Prove this.

If the eigenvalues of A can be arranged as

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{m-1}| > |\lambda_m| > 0.$$

Then, 0 is not an eigenvalue of A, and A^{-1} has eigenvalues $\frac{1}{\lambda_j}$ arranged as

$$|\lambda_m^{-1}| > |\lambda_{m-1}^{-1}| \geq \cdots \geq |\lambda_1^{-1}|.$$

Then, we can apply the power iteration to A^{-1} to approximate the smallest eigenvalue of A in magnitude!

Practical implementation: Bad idea to invert A and then define $z_k = A^{-1}x_{k-1}$ and normalise $x_k = \frac{z_k}{\|z_k\|_2}$ in order to generate the sequence $\{x_k\}_{k\in\mathbb{N}}$!

A better idea (at least for large matrices) is to factorise A (e.g. QR or SVD) and then solve

$$Az_{k+1} = x_k,$$

then normalise $x_{k+1} := \frac{z_{k+1}}{\|z_{k+1}\|_2}$.

Shifted inverse iteration

Given a diagonalizable matrix $A \in \mathbb{C}^{m \times m}$:

- Power iteration $\rightarrow$ approximate largest eigenvalue of A in magnitude.
- Inverse power iteration $\rightarrow$ approximate smallest eigenvalue of A in magitude.

What about those in between?

Suppose $\mu \in \mathbb{C}$ is not an eigenvalue of A. Then, $B := A - \mu I$ is invertible and the eigenvalues of B are $\{\lambda_1 - \mu, \lambda_2 - \mu, \ldots, \lambda_m - \mu\}$.

Suppose λ_J is an eigenvalue "closest" to μ, i.e.,

$$|\lambda_J - \mu| < |\lambda_i - \mu| \text{ for } i \neq J,$$

then we can use inverse iteration on $(A - \mu I)$ [equivalently power iteration on $(A - \mu I)^{-1}$] to find an approximation of $\lambda_J - \mu$.

This is the shifted inverse iteration that approximates the eigenvalue of A closest to the shift $\mu \in \mathbb{C}$.

Practical implementation: Factorise $A - \mu I$ and then solve

$$(A - \mu I) z_{k+1} = x_k, \quad x_{k+1} := \frac{z_{k+1}}{\|z_{k+1}\|_2}.$$

Gershgorin circle theorem I

How do we choose the shift $\mu \in \mathbb{C}$ in the shifted inverse iteration?

Theorem: Let $A \in \mathbb{C}^{m \times m}$. For $i \in \{1, \dots, m\}$, let $R_i = \sum_{j \neq i} |a_{ij}|$. Then, every eigenvalue of A lies within at least one of the so-called Gershgorin discs $D(a_{ii}, R_i)$, where

$$D(a_{ii}, R_i) = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i\}.$$

Heuristically: If off-diagonal entries of A have small norms, then the eigenvalues of A cannot be too "far from" the main diagonal entries of A.

Gershgorin circle theorem II

Theorem: Let $A \in \mathbb{C}^{m\times m}$. For $i \in \{1, \ldots, m\}$, let $R_i = \sum_{j\neq i} |a_{ij}|$. Then, every eigenvalue of A lies within at least one of the so-called Gershgorin discs $D(a_{ii}, R_i)$, where

$$D(a_{ii}, R_i) = \{z \in \mathbb{C} \,:\, |z - a_{ii}| \leq R_i\}.$$

Proof: (1) Let λ be an eigenvalue of A, and choose eigenvector x normalized so that $\|x\|_\infty = 1$. Let $i \in \{1, \ldots, m\}$ be the index for which $|x_i| = 1$.

(2) Since $Ax = \lambda x$, we have

$$\lambda x_i = \sum_{j=1}^{m} a_{ij}x_j \quad \Rightarrow \quad (\lambda - a_{ii})x_i = \sum_{j\neq i} a_{ij}x_j.$$

(3) Take absolute values, and use $|x_j| \leq 1 = |x_i|$ to get

$$|\lambda - a_{ii}| \leq \left|\sum_{j\neq i} a_{ij}x_j\right| \leq \sum_{j\neq i} |a_{ij}| = R_i.$$

$\square$

Gershgorin's circle theorem III

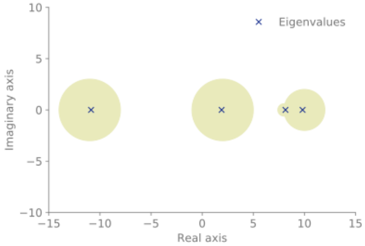
Note: it is possible that one disc can contain more than one eigenvalue.
Example

$$A = \begin{pmatrix} 10 & -1 & 0 & 1 \\ 0.2 & 8 & 0.2 & 0.2 \\ 1 & 1 & 2 & 1 \\ -1 & -1 & -1 & -11 \end{pmatrix}$$

Gershgorin's theorem says each eigenvalue of A are contained in the following four discs:

$$D(10,2), \quad D(8,0.6), \quad D(2,3), \quad D(-11,3)$$

The eigenvalues are 9.8218, 8.1478, 1.8995, -10.86.



Class exercise

1. Let $A \in \mathbb{C}^{m\times m}$ be diagonalizable with eigenvalues

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

   where $|\lambda_2|/|\lambda_1|$ is close to 1. Write down the algorithm (i.e., $\{y_k\}_{k\in\mathbb{N}}$) for the shifted power iteration for A with shift μ, and deduce the convergence rate of the shifted power iteration. What values of μ should you choose to improve the slow convergence of the power iteration?

2. The Rayleigh quotient of a non-zero vector $x \in \mathbb{C}^m$ and a matrix $A \in \mathbb{C}^{m\times m}$ is

$$r(A, x) = \frac{x^* A x}{x^* x}.$$

   - Show that if x is an eigenvector of A, then $r(A, x)$ is the corresponding eigenvalue.
   - Show that the partial derivative of r with respect to x_j is

$$\frac{\partial r(A, x)}{\partial x_j} = \frac{2}{x^* x}(Ax - r(A, x)x)_j.$$

   - Deduce that eigenvectors of A satisfies $\nabla_x r(A, x) = 0$.

Two phase eigenvalue computation I

Most general purpose eigenvalue algorithms used today employs the Schur factorization $A = QTQ^*$.

We apply similarity transformations $X \mapsto Q_j^* X Q_j$ to A with unitary matrices, so that the sequence $(B_i)_{i \geq 1}$ defined as

$$B_i = Q_i^* B_{i-1} Q_i, \quad B_1 = Q_1^* A Q_1$$

eventually converges to an upper triangular matrix T as $i \to \infty$.

The basic idea of the two phase eigenvalue computation is:

- Phase 1: Transform A into upper Hessenberg form, i.e., all entries below first subdiagonal are zero [$a_{ij} = 0$ for $i > j + 1$]. This can be done in a finite number of steps.
- Phase 2: Generate a sequence of upper Hessenberg matrices that converges to an upper triangular matrix. This is an iterative process.

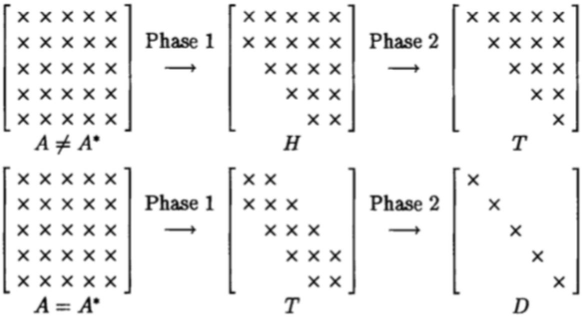
If A is hermitian, then we get tridiagonal matrices instead.

Two phase eigenvalue computation II

- Phase 1: Transform A into upper Hessenberg form, i.e., all entries below first subdiagonal are zero [$a_{ij} = 0$ for $i > j + 1$]. This can be done in a finite number of steps.
- Phase 2: Generate a sequence of upper Hessenberg matrices that converges to an upper triangular matrix. This is an iterative process.

If A is hermitian, then we get tridiagonal matrices instead.

Schematically:



Phase 1 – Reduction to upper Hessenberg form

Recall the Householder reflections that creates zeros below the first entry

$$x = \begin{bmatrix} \times \\ \times \\ \times \\ \vdots \\ \times \end{bmatrix} \quad \xrightarrow{F} \quad Fx = \begin{bmatrix} \|x\| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\|e_1.$$

So, one idea is to repeatedly use (appropriate) Householder reflections to introduce zeros below the main diagonal.

This turns out to be a bad idea! Schemtically, the first Householder reflector Q_1^* multiplied on the left of A will change all rows of A.

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \quad \xrightarrow{Q_1^* \cdot} \quad \underset{Q_1^* A}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}$$

Reduction to upper Hessenberg form II

Schemtically, the first Householder reflector Q_1^* multiplied on the left of A will change all rows of A.

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1^* \cdot} \underset{Q_1^* A}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}$$

To complete the similarity transformation, we have to multiply on the right by Q_1:

$$\underset{Q_1^* A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}} \xrightarrow{\cdot Q_1} \underset{Q_1^* A Q_1}{\begin{bmatrix} \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}$$

Therefore, all the zeros created before are now lost!

Reduction to upper Hessenberg form III

A better idea is to be less ambitious and aim for a Hessenberg form. Let Q_1^* be a Householder reflection that leaves the first row unchanged. Then, Q_1^* mutiplied on the left of A introduce zeros in row 3 and onwards of the first column.

$$
\underset{A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}
\xrightarrow{Q_1^* \cdot}
\underset{Q_1^* A}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
\xrightarrow{\cdot Q_1}
\underset{Q_1^* A Q_1}{\begin{bmatrix} \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
$$

When multiplying $Q_1^* A$ with Q_1 on the right, the first column is unchanged (by design), so the zeros we created are preserved.

The second Householder reflector Q_2^* would leave the first and second rows unchanged.

$$
\underset{Q_1^* A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix}}
\xrightarrow{Q_2^* \cdot}
\underset{Q_2^* Q_1^* A Q_1}{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \mathbf{0} & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
\xrightarrow{\cdot Q_2}
\underset{Q_2^* Q_1^* A Q_1 Q_2}{\begin{bmatrix} \times & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ \times & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & \times & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \\ & & \boldsymbol{\times} & \boldsymbol{\times} & \boldsymbol{\times} \end{bmatrix}}
$$

This process terminates after a total of $m-2$ steps, leading to

$$\underbrace{Q_{m-2}^* \cdots Q_1^*}_{Q^*} \, A \, \underbrace{Q_1 \cdots Q_{m-2}}_{Q} = H,$$

Reduction to upper Hessenberg form IV

Going back to the slide Finding the unitary matrices. We want Q_1^* to leave the first row unchanged. Meaning

$$Q_1^* = \begin{pmatrix} 1 & 0 \\ 0 & F \end{pmatrix}$$

where $F \in \mathbb{C}^{(m-1)\times(m-1)}$ is unitary.

The procedure is:

- Set $x = (a_{21}, \dots, a_{m1})^\top \in \mathbb{C}^{m-1}$ as the first column of A without the first entry a_{11}.
- Construct vector[3]

  $$v_1 = \mathrm{sign}(a_{21})\|x\|e_1 + x,$$

  where $e_1 = (1, 0, \dots, 0)^\top \in \mathbb{C}^{m-1}$.
- Construct Householder reflector

  $$F_1 = I - \frac{2v_1v_1^*}{\|v_1\|^2} \in \mathbb{C}^{(m-1)\times(m-1)}$$

  and set $F = F_1$.

[3] typo in previous versions, the sign

Reduction to upper Hessenberg form V

In the second step, we want Q_2^* to leave the first and second rows unchanged. Meaning

$$Q_2^* = \begin{pmatrix} I_{2\times 2} & 0 \\ 0 & F_2 \end{pmatrix}$$

where $F_2 \in \mathbb{C}^{(m-2)\times(m-2)}$ is unitary.

The procedure is:

- Set $x = (a_{32}, \dots, a_{m2})^\top \in \mathbb{C}^{m-2}$ as the second column of $Q_1^* A Q_1$ without the first and second entries.
- Construct vector[4]

  $$v_2 = \mathrm{sign}(a_{32})\|x\|e_1 + x,$$

  where $e_1 = (1, 0, \dots, 0)^\top \in \mathbb{C}^{m-2}$.
- Construct Householder reflector

  $$F_2 = I - \frac{2 v_2 v_2^*}{\|v_2\|^2} \in \mathbb{C}^{(m-2)\times(m-2)}.$$

and so on ...

[4] typo on previous versions, the sign

Phase 2 – Iterative algorithms

After transforming A into upper Hessenberg form (or tridiagonal form if A is hermitian), we now consider methods to approximate eigenvalues and eigenvectors.

The first is the Rayleigh quotient iteration derived from the shifted inverse iteration. Applied to hermitian matrices.

The second is an algorithm based on QR factorization. Applied to Hessenberg matrices.

Both are able to compute all eigenvalues and eigenvectors of the matrix.

Rayleigh quotient iteration

For fixed matrix $A \in \mathbb{C}^{m \times m}$, the Rayleigh quotient of a vector $x \in \mathbb{C}^m$ is

$$r(x) = \frac{x^* A x}{\|x\|_2^2}.$$

Related problem: Given $x \in \mathbb{C}^m$, find a scalar $\alpha \in \mathbb{C}$ "acting most like an eigenvalue" for x, in the sense that

$$\|Ax - \alpha x\|_2 \leq \|Ax - \beta x\|_2 \text{ for all } \beta \in \mathbb{C}.$$

Viewing x as a matrix in $\mathbb{C}^{m \times 1}$, the solution to the least squares problem

$$x\alpha = Ax$$

is (recall the normal equation)

$$\alpha = (x^* x)^{-1} x^* (Ax) = \frac{x^* A x}{x^* x}.$$

I.e., the Rayleigh quotient is the solution to the least squares problem.

In Class exercise we have shown

- if x is an eigenvector of A, then $r(x)$ is the corresponding eigenvalue.

Therefore, for given arbitrary $x \in \mathbb{C}^m$, the scalar $r(x)$ is a natural eigenvalue estimate.

Rayleigh quotient iteration II

So far, the Rayleigh quotient gives

$$\text{approx. eigenvector } x \quad \longrightarrow \quad \text{approx. eigenvalue } r(x).$$

Is there an algorithm that gives the reverse?

Yes! The shifted inverse iteration: For given $\mu \in \mathbb{C}$ and initial vector $x \in \mathbb{C}^m$, we generate a sequence that approximates the eigenvector associated to the eigenvalue of A closest to μ. I.e.,

$$\text{approx. eigenvalue } \mu \quad \longrightarrow \quad \text{approx. eigenvector } x.$$

The Rayleigh quotient iteration is simply to combine these two methods:

1. Initialise with vector $x_0 \in \mathbb{C}^m$ with $\|x_0\|_2 = 1$.
2. Compute Rayleigh quotient $r_0 = x_0^* A x_0$.
3. for $k = 1, 2, \cdots$
    - Solve $(A - rI)w = x_{k-1}$ for w (Inverse iteration).
    - Set $x_k = w / \|w\|_2$ (normalize).
    - Set $r_k = x_k^* A x_k$ (Rayleigh quotient).

Heuristically, in step k, we use the Inverse iteration with $\mu = r$ (previous Rayleigh quotient) to output an approximate eigenvector x_k, and then use this to compute a better approximate of the eigenvalue.

Rayleigh quotient iteration III

Why is this method so spectacular?

Theorem: The Rayleigh quotient iteration generates a sequence of $(r_k, x_k)_{k\in\mathbb{N}}$ such that when it converges to an eigenpair (λ_J, v_J) of A, the convergence is cubic, i.e.,

$$|r_k - \lambda_J| = \mathcal{O}(|r_{k-1} - \lambda_J|^3), \quad \|x_k - (\pm v_J)\|_2 = \mathcal{O}(\|x_{k-1} - (\pm v_J)\|_2^3),$$

where the $\pm$ signs are not necessarily the same on the two sides.

This means that the error at the k-th step is roughly the error at the $(k-1)$-th step raised to the third power.

Example: Consider

$$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 4 \end{pmatrix}.$$

Set $x_0 = (1,1,1)^\top/\sqrt{3}$. When the Rayleigh quotient iteration is applied to A, we get the following first three iterations

$$r_0 = 5, \quad r_1 = 5.2131\ldots, \quad r_2 = 5.214319743184\ldots.$$

The actual value of the eigenvalue corresponding to the eigenvector closest to x_0 is $\lambda = 5.214319743377$.

Rayleigh quotient iteration IV

Proof: (1) Suppose (λ, v) is an eigenpair of A. Using a Taylor expansion

$$\begin{aligned} r(x) &= r(v) + \nabla r(v)^*(x-v) + \tfrac{1}{2}(x-v)^* \nabla^2(r(x))(x-v) + \mathcal{O}(\|x-v\|_2^3) \\ &= r(v) + \tfrac{1}{2}(x-v)^* \nabla^2(r(x))(x-v) + \mathcal{O}(\|x-v\|_2^3) \end{aligned}$$

since in Class exercise $\nabla r(v) = 0$ for an eigenvector v. Hence, for $\lambda := r(v)$,

$$|r(x) - \lambda| = \mathcal{O}(\|x-v\|_2^2).$$

(2) Therefore, if $\|x_k - (\pm v_J)\|_2 = \mathcal{O}(\varepsilon)$, the Rayleigh quotient yields an estimate for the approximate eigenvalue r_k with $|r_k - \lambda_J| = \mathcal{O}(\varepsilon^2)$.

(3) For the Power iteration (see slide Power iteration III), there is an estimate

$$\|x_k - v_J\|_2 = \mathcal{O}\left(\left| \frac{\lambda_2}{\lambda_1} \right|^k \right),$$

where λ_1 and λ_2 are the largest and 2nd largest eigenvalue of A. So for the shifted inverse iteration applied to $A - r_k I$, the eigenvalues of $(A - r_k I)^{-1}$ are

$$\frac{1}{\lambda_1 - r_k}, \quad \frac{1}{\lambda_2 - r_k}, \quad \dots, \quad \frac{1}{\lambda_m - r_k}.$$

Rayleigh quotient iteration V

(4) If λ_J is the closest eigenvalue to r_k and λ_K is the 2nd closest, then we get the estimate for an iteration of the shifted inverse iteration:

$$\|x_{k+1} - (\pm v_J)\|_2 = \mathcal{O}\Big(\Big| \frac{\lambda_J - r_k}{\lambda_K - r_k} \Big|^{k+1} \Big) = \mathcal{O}\Big(\Big| \frac{\lambda_J - r_k}{\lambda_K - r_k} \Big|^{k} \Big| \frac{\lambda_J - r_k}{\lambda_K - r_k} \Big| \Big)$$

(5) Since we assumed the Rayleigh quotient iteration is convergent, this means

$$\Big| \frac{\lambda_J - r_k}{\lambda_K - r_k} \Big| = \Big| \frac{\lambda_J - r_k}{\lambda_K - \lambda_J + \lambda_J - r_k} \Big| = \mathcal{O}\Big(\Big| \frac{\lambda_J - r_k}{\lambda_K - \lambda_J} \Big| \Big) = \mathcal{O}(|\lambda_J - r_k|).$$

as $\lambda_J \neq \lambda_K$. Then, (see previous slide)

$$\|x_{k+1} - (\pm v_J)\|_2 = \mathcal{O}\big(\|x_k - (\pm v_J)\|_2 |\lambda_J - r_k|\big) = \mathcal{O}(\|x_k - (\pm v_J)\|_2^3),$$

and

$$\begin{aligned} |r_{k+1} - \lambda_J| &= \mathcal{O}\big(\|x_{k+1} - (\pm v_J)\|_2^2\big) = \mathcal{O}\big(\|x_k - (\pm v_J)\|_2^2 |\lambda_J - r_k|^2\big) \\ &= \mathcal{O}\big(|\lambda_J - r_k|^3\big). \end{aligned}$$

$\square$

A more detailed proof can be found in the book of J.W. Demmel, Applied Numerical Linear Algebra, Section 5.3.2

Class exercise

1. Let $A \in \mathbb{C}^{m \times m}$ be given, not necessarily hermitian. Show that a number $z \in \mathbb{C}$ is a Rayleigh quotient of A if and only if it is a diagonal entry of $Q^* A Q$ for some unitary matrix Q.

2. Use the Rayleigh quotient iteration to compute an eigenpair for the matrix

$$A = \begin{pmatrix} 5 & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & 7 \end{pmatrix}$$

   with $x_0 = (1, 0, 0)^\top$.

QR algorithm

The basic idea: Starting from the original matrix $A_0 := A$, we generate a sequence of matrices $(A_k)_{k\in\mathbb{N}}$ with the QR decomposition. Suppose at step k we have

$$Q_k R_k = A_k,$$

where Q_k is unitary and R_k is upper triangular. We define

$$A_{k+1} = R_k Q_k$$

(just swapping the order of multiplication). Then,

$$A_{k+1} = R_k Q_k = Q_k^* Q_k (R_k Q_k) = Q_k^* A_k Q_k.$$

I.e., A_k and A_{k+1} are similar, and so they have the same eigenvalues.

Under certain conditions, the sequence $(A_k)_{k\in\mathbb{N}}$ conveges to the Schur form of A, i.e., the upper triangular matrix U in the Schur factorization of $A = Q^* U Q$. The eigenvalues are listed on the main diagonal of U.

In practice, matrix A is brought into upper Hessenberg form first, and then the QR algorithm is applied.

Simultaneous iteration

We will relate the QR algorithm to another method called simultaneous iteration. Recalling that previous methods (Power iteration, Inverse iteration, etc.) can only compute 1 eigenvalue at a time.

Is there a way to compute more eigenvalues simultaneously?

If $A \in \mathbb{C}^{m \times m}$ has m LI eigenvectors $\{v_1, \dots, v_m\}$, the Power iteration starts with an initial vector x_0 that can be written as a linear combination

$$x_0 = a_1 v_1 + \cdots + a_m v_m.$$

Eigenvectors not orthogonal to x_0 will have a chance to be found by the Power iteration. E.g., in the original we had to assume that $a_1 \neq 0$ in order to find v_1.

Therefore, we should try applying the Power iteration to several different starting vectors, each orthogonal to each other, in order to find different eigenvalues.

Simultaneous iteration II

The idea is as follows: For $n \leq m$, given a set of n LI vectors $\{x_1^{(0)}, \dots, x_n^{(0)}\}$, we consider the iteration

$$A^k x_1^{(0)}, \quad A^k x_2^{(0)}, \quad \dots, \quad A^k x_n^{(0)}.$$

In matrix notation, we define $X^{(0)} \in \mathbb{C}^{m \times n}$ to be the matrix

$$X^{(0)} = \begin{pmatrix} | & & | \\ x_1^{(0)} & \cdots & x_n^{(0)} \\ | & & | \end{pmatrix}$$

and $X^{(k)}$ to be the result after k applications of A:

$$X^{(k)} = A^k X^{(0)} = \begin{pmatrix} | & & | \\ x_1^{(k)} & \cdots & x_n^{(k)} \\ | & & | \end{pmatrix}$$

This can be viewed as the Power iteration applied to all the vectors $\{x_1^{(0)}, \dots, x_n^{(0)}\}$ at once.

We expect as $k \to \infty$, the columns of $X^{(k)}$ converges to scalar copies of v_1, the unit eigenvector corresponding to the largest eigenvalue of A in magnitude.

Simultaneous iteration III

So far, there is no new information! as the n eigenvectors can possibly be in the same direction.

However, in the original Power iteration there is a step of normalization. For the multi-vector version, the analogue is to obtain an orthonormal set of eigenvector estimates during each iteration.

This forces the eigenvector approximations to be orthogonal at all times, and is done by computing the QR factorization of $X^{(k)}$.

Heuristically: if $x_1^{(k)}$ converges to v_1, then as $x_2^{(k)}$ is orthogonal to $x_1^{(k)}$, it should converge to v_2, the eigenvector corresponding to the second largest eigenvalue in magnitude. Then, $x_3^{(k)}$ being orthogonal to $\{x_1^{(k)}, x_2^{(k)}\}$ should converge to v_3, etc....

Recall the reduced QR factorization of a matrix $A \in \mathbb{C}^{m\times n}$ is

$$A = \hat{Q}\hat{R}$$

where $\hat{R} \in \mathbb{C}^{n\times n}$ is upper triangular with non-zero diagonal, and $\hat{Q} \in \mathbb{C}^{m\times n}$ with orthonormal columns.

Simultaneous iteration IV

We now describe the Simultaneous iteration method:

1. Pick a starting LI set $\{x_1^{(0)}, \dots, x_n^{(0)}\}$ with $n \leq m$.
2. Build matrix $X^{(0)}$ with columns $x_1^{(0)}, \dots, x_n^{(0)}$.
3. Obtain reduced QR factorization $\hat{Q}^{(0)}\hat{R}^{(0)} = X^{(0)}$.

   For $k = 1, 2, \dots$
   - Set $W^{(k)} = A\hat{Q}^{(k-1)}$.
   - Obtain reduced QR factorization $\hat{Q}^{(k)}\hat{R}^{(k)} = W^{(k)}$

Under suitable conditions, the columns of $\hat{Q}^{(k)}$ will converge to $\pm v_1, \pm v_2, \dots, \pm v_n$, the eigenvectors corresponding to the n largest eigenvalues of A in magnitude.

An informal explanation: The columns $\{q_1^{(0)}, \dots, q_n^{(0)}\}$ of $\hat{Q}^{(0)}$ is an orthonormalization of the columns of $X^{(0)}$. Then, $W^{(1)}$ is the action of the matrix A on these orthonormal columns, and the reduced QR factorization yields the next set of approximate eigenvectors as columns of $\hat{Q}^{(1)}$.

Simultaneous iteration ⇔ QR algorithm

It turns out that the QR algorithm is equivalent to the simultaneous iteration with $n = m$. In this case we use full QR factorizations instead, and with initial orthonormal matrix $\underline{Q}^{(0)} = I_{m\times m}$.

The Simultaneous iteration reads:

- $\underline{Q}^{(0)} = I_{m\times m}$,
- $W^{(k)} = A\underline{Q}^{(k-1)}$,
- $\underline{Q}^{(k)}R^{(k)} = \underline{W}^{(k)}$,

and we set $\underline{A}^{(k)} = (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)}$.

The QR algorithm reads:

- $A^{(0)} = A$,
- $Q^{(k)}R^{(k)} = A^{(k-1)}$,
- $A^{(k)} = R^{(k)}Q^{(k)}$,

and we set $\underline{Q}^{(k)} = Q^{(1)}\cdots Q^{(k)}$.

Introducing an additional matrix

$$\underline{R}^{(k)} = R^{(k)}R^{(k-1)}\cdots R^{(1)} = R^{(k)}\underline{R}^{(k-1)}.$$

Simultaneous iteration $\Leftrightarrow$ QR algorithm II

Theorem: The two processes are equivalent. They both generate identical sequences of matrices $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$ and $A^{(k)}$. Moreover, it holds that the k-th power of A has the QR factorization

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

and the k-th iteration has the formula

$$A^{(k)} = (\underline{Q}^{(k)})^\top A \underline{Q}^{(k)}$$

Proof by induction: Case $k = 0$. By design $\underline{Q}^{(0)} = I$, and by definition

$$A^0 = I \quad \Rightarrow \quad \underline{R}^{(0)} = (\underline{Q}^{(0)})^\top A^0 = I,$$

and

$$A^{(0)} = (\underline{Q}^{(0)})^\top A \underline{Q}^{(0)} = A.$$

Simultaneous iteration $\Leftrightarrow$ QR algorithm III

Theorem: The two processes are equivalent. They both generate identical sequences of matrices $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$ and $A^{(k)}$. Moreover, it holds that the k-th power of A has the QR factorization

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

and the k-th iteration has the formula

$$A^{(k)} = (\underline{Q}^{(k)})^\top A \underline{Q}^{(k)}$$

Consider $k \geq 1$: For Simultaneous iteration, the formula for $A^{(k)}$ is by definition.

Meanwhile, by induction and the formula $\underline{Q}^{(k)} R^{(k)} = A\underline{Q}^{(k-1)}$, we have

$$A^k = AA^{k-1} = A\underline{Q}^{(k-1)}\underline{R}^{(k-1)} = \underline{Q}^{(k)} R^{(k)} \underline{R}^{(k-1)} = \underline{Q}^{(k)}\underline{R}^{(k-1)}.$$

Simultaneous iteration $\Leftrightarrow$ QR algorithm IV

Theorem: The two processes are equivalent. They both generate identical sequences of matrices $\underline{R}^{(k)}$, $\underline{Q}^{(k)}$ and $A^{(k)}$. Moreover, it holds that the k-th power of A has the QR factorization

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

and the k-th iteration has the formula

$$A^{(k)} = (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)}$$

Consider $k \geq 1$: For QR algorithm, using that $R^{(k)} = (Q^{(k)})^\top A^{(k-1)}$:

$$A^{(k)} = R^{(k)}Q^{(k)} = (Q^{(k)})^\top A^{(k-1)}Q^{(k)} = (Q^{(k)})^\top \big((Q^{(k-1)})^\top A^{(k-2)}Q^{(k-1)}\big)Q^{(k)}$$
$$= \cdots = \underline{Q}^{(k)}A\underline{Q}^{(k)}.$$

Meanwhile, by induction hypothesis, i.e., $A^{k-1} = \underline{Q}^{(k-1)}\underline{R}^{(k-1)}$ and $A^{(k-1)} = (\underline{Q}^{(k-1)})^\top A\underline{Q}^{(k-1)}$, it holds

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

$\square$

Convergence of the QR algorithm

The QR algorithm takes an initial matrix A (real and symmetric) and outputs a sequence $\{A^{(k)}\}_{k\in\mathbb{N}}$ along with QR-type factors $\{Q^{(k)}, R^{(k)}\}_{k\in\mathbb{N}}$.

By previous theorem, we have the formula

$$A^k = \underline{Q}^{(k)}\underline{R}^{(k)}, \quad A^{(k)} = (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)},$$
$$\text{where } \underline{Q}^{(k)} = Q^{(1)}\cdots Q^{(k)}, \quad \underline{R}^{(k)} = R^{(k)}\cdots R^{(1)}.$$

Theorem: Let the QR algorithm be applied to a real symmetric matrix A whose eigenvalues satisfy $|\lambda_1| > \cdots > |\lambda_m|$, and whose corresponding eigenvector matrix Q has all nonzero leading principal minors. Then,

$$A^{(k)} \to \mathrm{diag}(\lambda_1, \ldots, \lambda_m) =: \Lambda \text{ as } k \to \infty,$$

and $\underline{Q}^{(k)}$ (adjusting the signs of its columns as necessary) converges to Q.

Recall: the k-th leading principal minor of a matrix $A \in \mathbb{C}^{m\times m}$ is the determinant of the upper-left $k \times k$ submatrix.

Note: A invertible $\not\Rightarrow$ all principal minors nonzero, e.g. $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ is invertible, but its first principal minor is $\{0\}$.

Convergence of the QR algorithm II

Theorem: Let the QR algorithm be applied to a real symmetric matrix A whose eigenvalues satisfy $|\lambda_1| > \cdots > |\lambda_m|$, and whose corresponding eigenvector matrix Q has all nonzero leading principal minors. Then,

$$A^{(k)} \to \mathrm{diag}(\lambda_1, \ldots, \lambda_m) =: \Lambda \text{ as } k \to \infty,$$

and $\underline{Q}^{(k)}$ (adjusting the signs of its columns as necessary) converges to Q.

Proof ingredients:

- The eigenvalue decomposition of a real symmetric matrix A is $A = Q\Lambda Q^\top$, with orthogonal matrix Q (i.e., $Q^\top = Q^{-1}$) and diagonal Λ;
- Uniqueness of QR factorization: If $A \in \mathbb{R}^{m\times m}$ have LI columns, and $A = Q_1 R_1 = Q_2 R_2$ are two QR factorizations, then $Q_1 = Q_2$ and $R_1 = R_2$ (Exercise).
- If an invertible matrix A has all nonzero leading principal minors, then it admits an LU factorization, i.e., $A = LU$ where U is upper triangular and L is lower triangular with 1s on the main diagonal (aka unit lower triangular).

Convergence of the QR algorithm III

Proof: (1) Let $Q^\top = LU$ be the LU factorization of $Q^\top$. Then, for any $k \in \mathbb{N}$,

$$A^k = Q\Lambda^k Q^\top = Q\Lambda^k LU.$$

Hence,

$$Q\Lambda^k L\Lambda^{-k} = A^k U^{-1}\Lambda^{-k} = \underline{Q}^{(k)}\underline{R}^{(k)}U^{-1}\Lambda^{-k}.$$

(2) The matrix $\Lambda^k L \Lambda^{-k}$ satisfies

$$(\Lambda^k L\Lambda^{-k})_{ij} = \begin{cases} l_{ij}(\lambda_j/\lambda_i)^k & i > j, \\ 1 & i = j, \\ 0 & i < j. \end{cases}$$

Since $|\lambda_i| > |\lambda_j|$ if $j < i$, we see

$$\Lambda^k L\Lambda^{-k} \to I_{m\times m}, \quad \underline{Q}^{(k)}\underline{R}^{(k)}U^{-1}\Lambda^{-k} \to Q$$

as $k \to \infty$.

(3) By uniqueness of QR factorization,

$$\underline{Q}^{(k)} \to Q, \quad \underline{R}^{(k)}U^{-1}\Lambda^{-k} \to I_{m\times m} \text{ as } k \to \infty.$$

Then,

$$A^{(k)} := (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)} = (\underline{Q}^{(k)})^\top Q\Lambda Q^\top \underline{Q}^{(k)} \to Q^\top Q\Lambda QQ^\top = \Lambda \text{ as } k \to \infty.$$

$\square$

QR algorithm with shifts

A simple example where the QR algorithm "fails" :

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Eigenvalues are 1 and -1. The QR algorithm applied to this matrix gives

$$Q^{(k)} = A, \quad R^{(k)} = I \quad \Longrightarrow \quad A^{(k)} = A \text{ for all } k \in \mathbb{N}.$$

The QR algorithm stagnates and there is no convergence, obvious from the Theorem as we have

- $|\lambda_1| = |\lambda_2|$ where $\lambda_1 = 1$ and $\lambda_2 = -1$.
- First principal minor $\{0\}$ is zero.

To fix things, we introduce the QR algorithm with shifts, and call the previous algorithm the QR algorithm without shifts / unshifted QR algorithm.

QR algorithm with shifts II

Assume again $A \in \mathbb{R}^{m\times m}$ is symmetric.

The unshifted QR algorithm is the simultaneous iteration applied to the identity matrix $I_{m\times m}$, and the first column evolves according to the Power iteration.

A dual observation: The unshifted QR algorithm is also equivalent to a simultaneous inverse iteration applied to a "flipped" identity matrix P

$$P = \begin{pmatrix} & & & & 1 \\ & & & 1 & \\ & & 1 & & \\ & \dots & & & \\ 1 & & & & \end{pmatrix}$$

To be more precise. We recall that the k-th power of A has the QR factorization

$$A^k = \underline{Q}^{(k)}\underline{R}^{(k)}, \quad \text{where } \underline{Q}^{(k)} = Q^{(1)}\cdots Q^{(k)}, \quad \underline{R}^{(k)} = R^{(k)}\cdots R^{(1)}.$$

Inverting this formula and using that A^{-1} is symmetric:

$$A^{-k} = (\underline{R}^{(k)})^{-1}(\underline{Q}^{(k)})^{\top} = \underline{Q}^{(k)}(\underline{R}^{(k)})^{-\top} = (A^{-k})^{\top}.$$

QR algorithm with shifts III

Using $P^2 = I_{m\times m}$, we have

$$A^{-k}P = \underline{Q}^{(k)}(\underline{R}^{(k)})^{-\top}P = \left(\underline{Q}^{(k)}P\right)\left(P(\underline{R}^{(k)})^{-\top}P\right)$$

Observe:

- The factor $\left(\underline{Q}^{(k)}P\right)$ is orthogonal, i.e.,

$$\left(\underline{Q}^{(k)}P\right)\left(\underline{Q}^{(k)}P\right)^{\top} = I_{m\times m}.$$

- The factor $\left(P(\underline{R}^{(k)})^{-\top}P\right)$ is upper triangular. Applying P on the right flips the matrix left-to-right, and applying P on the left flips the matrix top-to-bottom.

So we have a QR factorization of $A^{-k}P$. But we can interpret $A^{-k}P$ as the result after k applications of A^{-1} to the initial matrix P.

I.e., we are applying simultaneous iteration with matrix A^{-1} to initial matrix P.

Equivalently, we are applying simultaneous inverse iteration with matrix A to initial matrix P.

QR algorithm with shifts IV

Since the QR algorithm can be viewed as a simultaneous inverse iteration, we can use shifts to accelerate the performance.

The unshifted QR algorithm reads:

- $A^{(0)} = A$,
- $Q^{(k)}R^{(k)} = A^{(k-1)}$,
- $A^{(k)} = R^{(k)}Q^{(k)}$,

We simply introduce a shift $\mu^{(k)}$ as follows:

- $A^{(0)} = A$,
- $Q^{(k)}R^{(k)} = A^{(k-1)} - \mu^{(k)}I_{m\times m}$,
- $A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)}I_{m\times m}$,

What changed?

- We still have

$$\begin{aligned} A^{(k)} &= R^{(k)}Q^{(k)} + \mu^{(k)}I_{m\times m} = (Q^{(k)})^\top(A^{(k-1)} - \mu^{(k)}I_{m\times m})Q^{(k)} + \mu^{(k)}I_{m\times m} \\ &= (Q^{(k)})^\top A^{(k-1)}Q^{(k)} = \cdots \text{ by induction } \cdots = (\underline{Q}^{(k)})^\top A\underline{Q}^{(k)}. \end{aligned}$$

- But, we now have (by induction)

$$(A - \mu^{(k)}I_{m\times m})(A - \mu^{(k-1)}I_{m\times m})\cdots(A - \mu^{(1)}I_{m\times m}) = \underline{Q}^{(k)}\underline{R}^{(k)}.$$

QR algorithm with shifts V

The shifted QR algorithm is

- $A^{(0)} = A$,
- $Q^{(k)}R^{(k)} = A^{(k-1)} - \mu^{(k)} I_{m\times m}$,
- $A^{(k)} = R^{(k)}Q^{(k)} + \mu^{(k)} I_{m\times m}$,

where by an induction proof

$$A^{(k)} = (\underline{Q}^{(k)})^\top A \underline{Q}^{(k)}, \quad \prod_{j=1}^{k}(A - \mu^{(j)} I_{m\times m}) = \underline{Q}^{(k)}\underline{R}^{(k)}.$$

What are good choices for $\mu^{(j)}$? Rayleigh quotient $r(x) = \frac{x^\top A x}{\|x\|_2}$ which is the best approximation to an eigenvalue for the vector x.

One good choice is the Rayleigh quotient

$$\mu^{(k)} = \frac{(q_m^{(k)})^\top A q_m^{(k)}}{\|q_m^{(k)}\|_2} = (q_m^{(k)})^\top A q_m^{(k)},$$

where $q_m^{(k)}$ is the last column of $\underline{Q}^{(k)}$. Another formula for $\mu^{(k)}$ is

$$\mu^{(k)} = (q_m^{(k)})^\top A q_m^{(k)} = e_m^\top (\underline{Q}^{(k)})^\top A \underline{Q}^{(k)} e_m = e_m^\top A^{(k)} e_m = (A^{(k)})_{mm},$$

i.e., the (m, m)-th entry of $A^{(k)}$. The resulting algorithm is called Rayleigh quotient shifted QR algorithm.

Choice of shifts - Explanation

Suppose $A \in \mathbb{R}^{m\times m}$ has the form

$$A = \begin{pmatrix} \mathcal{A} & b \\ b^\top & c \end{pmatrix}$$

with $\mathcal{A} \in \mathbb{R}^{(m-1)\times(m-1)}$, $b \in \mathbb{R}^{m-1}$ and $c \in \mathbb{R}$. If entries of b are close to zero, then the standard basis vector e_m is nearly an eigenvector of A with c acting nearly as the eigenvalue.

Do one step of QR iteration and find orthogonal Q and upper triangular R such that

$$QR = A - cI_{m\times m}.$$

Symmetry of A implies

$$A - cI_{m\times m} = R^\top Q^\top \quad \Longrightarrow \quad Q = (A - cI_{m\times m})^{-1} R^\top.$$

Looking at the last column, since $R^\top$ is lower triangular, we see that (with r_{mm} as the (m,m)-th entry of R)

$$q_m = r_{mm}(A - cI_{m\times m})^{-1} e_m.$$

This is one step of shifted inverse iteration applied to $r_{mm}e_m$!

So if we choose c as the Rayleigh quotient of q_m, i.e., $c = q_m^\top A q_m$, and do another step of the shifted QR iteration, we obtain a new orthogonal matrix $\tilde{Q}$ whose last column is an even better approximation to the eigenvector than q_m.

Computing SVD

Recall for $A \in \mathbb{C}^{m \times n}$ $(m \geq n)$, the (full) SVD of A is $A = U\Sigma V^*$, where

- $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ are unitary
- $\Sigma \in \mathbb{C}^{m \times n}$ contains the singular values of A in decreasing order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ on its diagonal.

Relation between singular values and eigenvalues?

$$(\sigma_i(A))^2 = \lambda_i(A^*A) \text{ for } i = 1, \dots, n.$$

Meaning? We can calculate the SVD of A using the following algorithm:

1. Form the matrix A^*A.
2. Compute the eigenvalue decomposition $A^*A = V\Lambda V^*$.
3. Let $\Sigma \in \mathbb{C}^{m \times n}$ be the nonnegative diagonal square root of Λ.
4. Solve $U\Sigma = AV$ for unitary U (e.g. via QR factorization).

Justification for step (3): If $A = U\Sigma V^*$, then

$$A^*A = V\Sigma^* U^* U \Sigma V^* = V\Sigma^*\Sigma V^* \quad \Longrightarrow \quad \Lambda = \Sigma^*\Sigma.$$

Computing SVD II

Unfortunately, the above algorithm is unstable, in the sense that small perturbations (e.g. due to rounding) of the matrix A can yield large errors in singular values.

Let us consider an alternative idea for square matrices $A \in \mathbb{C}^{m\times m}$, by building the $2m \times 2m$ hermitian matrix

$$H = \begin{pmatrix} 0 & A^* \\ A & 0 \end{pmatrix}.$$

If $A = U\Sigma V^*$, then $AV = U\Sigma$ and $A^*U = V\Sigma^* = V\Sigma$ (since the entries of Σ are nonnegative real numbers). This implies

$$H\begin{pmatrix} V & V \\ U & -U \end{pmatrix} = \begin{pmatrix} 0 & A^* \\ A & 0 \end{pmatrix}\begin{pmatrix} V & V \\ U & -U \end{pmatrix} = \begin{pmatrix} V & V \\ U & -U \end{pmatrix}\begin{pmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{pmatrix}$$

This is an eigenvalue decomposition for H! I.e., singular values of A can be extracted from eigenvalues of H, and the matrices U and V can be extracted from the eigenvectors of H.

New algorithm (more stable than the first one) is:

1. Form the hermitian matrix H.
2. Reduce H into a tridiagonal form (see slide Two phase eigenvalue computation) - Phase 1.
3. Apply Rayleigh quotient iteration to get eigenvalues and eigenvectors of H - Phase 2.

Class exercise

1. Apply the unshifted QR algorithm to the following matrix

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & 2 & 1 \end{pmatrix}$$

2. Apply the QR algorithm with shift to the following matrix

$$B = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

Summary of Part 1

Singular value decomposition

1. The SVD of a matrix $A = U\Sigma V^*$ exists for any matrix $A \in \mathbb{C}^{m\times n}$.
   - Find eigenvalues $\sigma_1^2 \geq \sigma_2^2 \geq \cdots \geq \sigma_n^2$ to A^*A.
   - Find orthonormal set of eigenvectors $\{v_1, \dots, v_n\}$ and build matrix V with these as columns.
   - Set $u_i = \frac{1}{\sigma_i} A v_i$ and (add arbitrary orthonormal vectors) to build matrix U.
   - Set Σ as the diagonal matrix with entries $\sigma_1, \dots, \sigma_n$.
2. SVD can be written as a sum of rank-one matrices

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^*, \quad \text{where } r = \text{rank}(A).$$

3. For large square matrices A, practical implementations of SVD can be done by
   - Reducing

$$H = \begin{pmatrix} 0 & A^* \\ A & 0 \end{pmatrix} \in \mathbb{C}^{2m\times 2m}.$$

     to tridiagonal form.
   - Apply Rayleigh quotient iteration to get eigenvalues and eigenvectors of H.
   - Extract U, V and Σ from the eigenvalue decomposition of H.

Summary of Part 1 - continued

QR factorization

1. The QR factorization of a matrix $A = QR$ exists for any matrix $A \in \mathbb{C}^{m \times n}$.
   - Apply the Gram–Schmidt orthonormalization process to the columns of A yields the reduced QR factorization.
   - Apply Householder reflections to obtain the full QR factorization.
2. Both methods rely heavily on the notion of orthogonal projections.
3. A square matrix P is a projector if $P^2 = P$, and it is orthogonal if and only if P is hermitian.
4. For any matrix $B \in \mathbb{C}^{m \times n}$, the orthogonal projector onto range(B) is $P = BB^*$.

Least squares problem

1. The least squares problem is to find the best vector $x \in \mathbb{C}^m$ such that for $A \in \mathbb{C}^{m \times n}$ of full rank ($m \geq n$) and $b \in \mathbb{C}^n$,

$$\|b - Ax\|_2 \leq \|b - Ay\|_2 \text{ for all } y \in \mathbb{C}^m.$$

2. The solution is given by the normal equation $x = (A^*A)^{-1}A^*b$.
3. When A is full rank, use reduced QR factorization on A to compute x.
4. If A is rank-deficient, then use reduced SVD on A.

Summary of Part 1 – continued

Eigenvalue revealing factorizations

1. Matrices $A, B \in \mathbb{C}^{m\times m}$ are similar if there is an invertible $X \in \mathbb{C}^{m\times m}$ such that $B = X^{-1}AX$.
2. Similar matrices share the same eigenvalues and their multiplicity.
3. $A \in \mathbb{C}^{m\times m}$ is non-defective if and only if it admits an eigenvalue decomposition $A = X\Lambda X^{-1}$ with X invertible (columns are eigenvectors) and Λ diagonal (with eigenvalues as entries).
4. The Schur factorization of $A \in \mathbb{C}^{m\times m}$ is $A = QTQ^*$ with Q unitary and T upper triangular. Eigenvalues of A appear on the main diagonal of T.
5. Every square matrix has a Schur factorization.

Eigenvalue algorithms - Two step approach

1. (a) Transform $A \in \mathbb{C}^{m\times m}$ into upper Hessenberg form $\tilde{A}$ (done in a finite no. of steps),
   (b) Generate a sequence of upper Hessenberg matrices $B_i = Q_i^* B_{i-1} Q_i$ with $B_1 = Q_1^* \tilde{A} Q_1$ which converge to an upper triangular matrix T (iterative process)
2. Need to modify the Householder reflections in a suitable way for step 1!
3. For step 2, the QR algorithm is used to obtain all eigenvalues and eigenvectors.
4. QR algorithm is equivalent to the simultaneous iteration (aka applying Power iteration to multiple initial vectors simultaneously).
5. Performance accelerated by using the QR algorithm with shifts.