Norms

- Properties of a valid norm:
  - Should be real & non-negative
  - $\|x\| = 0 \iff x = 0$
  - $\|\alpha x\| = \alpha \|x\|$
  - $\|x + y\| \le \|x\| + \|y\|$
- $|x^T y| \le \|x\|_2 \|y\|_2$ (Cauchy-Shwarz Inequality)
- $\|A\|_1 = \max_{1 \le i \le m} (\|a_j\|_1)$
- $\|A\|_\infty$ = max. row absolute sum
- If $A = uv^T$, $\|A\|_2 = \|u\|_2 \|v\|_2$ (when $x = \frac{v}{\|v\|}$)
- $\|AB\|_p \le \|A\|_p \|B\|_p$
- $\|A\|_F = \sqrt{\sum_{i=0} \sum_{j=0} |a_{ij}|^2} = \sqrt{tr(AA^T)} = \sqrt{tr(A^TA)} = \sqrt{\sum_{i=0}^{n} \|a_i\|^2}$

Conditioning & Stability

- $\kappa(A) = \frac{\sigma_1}{\sigma_m} = \sqrt{\frac{\lambda_{max}(A^TA)}{\lambda_{min}(A^TA)}}$
- $\kappa^R(A) = \|A\| \|A^+\|$, where $\|A^+\| = (A^TA)^{-1}A^T$
- Backwards stable $\Rightarrow$ Stable $\Rightarrow$ Forward stable; Backwards stable $\nLeftarrow$ Stable $\nLeftarrow$ Forward stable

Backward stability: $\tilde{f}(x) = f(x + \delta x)$

- For a backward stable algo, $f(\tilde{x}) = \tilde{f}(x)$

Forward stability: $\|\tilde{f}(x) - f(x)\| = O(\varepsilon_m)$

Principal Component Analysis

1) Make the matrix 0-centered by subtracting the mean of the coln. from each element in the coln.
2) Variance of the coln. $= \frac{E(X^2) - \cancel{(E(X))^2}}{m} = \frac{\|a_j\|_2^2}{m}$ ($E(X) = 0$)
3) Hence, total variance, $T = \|a_1\|_2^2 + \dots + \|a_n\|^2$
   $= \|A\|_F^2$
   $= \sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2$

Biggest contributor to the variance

- To get the direction with highest variance, we need to find a vector $\hat{w}_1$ s.t.: $t_1 = A\hat{w}_1$, $\|t_1\|_2$ is maximised. The solution is $\hat{w}_1 = v_1$.

$$t_1 = Av_1 = \sum_{i=1}^{n} u_i \sigma_i v_i^T v_1 \qquad \left[v_i^T v_1 = \begin{cases} 0, & i \ne 1 \\ 1, & i = 1 \end{cases}\right]$$
$$= \sigma_1 u_1$$

- Similarly, $t_2 = \sigma_2 u_2$, where t_2 is the direction along second highest variance

Symmetric +ve Definite Matrix (SPD)

- Properties
  - $A = A^T$
  - $(x, Ay) = (y, Ax) \quad \forall x, y \in R^m$
  - $x^T A x > 0, \quad \forall x \in R^m$
- If A is S.P.D. & $X \in R^{m \times n}$ is full rank, then $X^T A X$ is also S.P.D.
- All eigenvalues are +ve for any S.P.D.

Cholesky Decomposition

- If A is S.P.D, A can be decomposed s.t. $A = R^T R$
- $n^3/3$ FLOPs

Projection

- Projector matrix P, must be:
  - $P^2 = P$ (idempotent) → $Pv - v \in Null(P)$
  - Square matrix → Rank-deficient
  - $Px = x \;\forall x \in Range(P)$
- Orthogonal projector also fulfils $P = P^T$ (symmetric)
- Let us have $Ax = b$ has no soln., we can solve $A\hat{x} = Pb$ instead. P projects b onto the coln. space of A, which makes this eqn. solvable.
  Hence, $P = A(A^TA)^{-1}A^T$
- If P is an orthogonal projector, so is $(I - P)$
- Projection & SVD: If $A = U\Sigma V^T$, U forms a orthonormal basis of A.
  Hence, the projection matrix corresponding to A, $P = Q(\cancel{Q^TQ})^{-1}Q^T = QQ^T$ ($Q^TQ = I$)

Eigen Decomposition

- Geometric multiplicity of λ: No. of L.I. eigenvectors associated with an eigenvalue λ.
- If λ corresponds to two eigenvectors v_1 & v_2, any vector $x = \alpha_1 v_1 + \alpha_2 v_2$ will be an eigenvector of A

QR Factorisation

- Gram-Schmidt orthogonalisation:
  $v_j = a_j - (q_1^T a_j) q_1 - \dots - (q_{j-1}^T a_j) q_{j-1}$
  $q_j = \frac{v_j}{\|v_j\|_2}$

Hence $q_n = \frac{a_n - \sum_{i=0}^{n-1} r_{in} q_i}{r_{nn}}$, where $r_{ij} = \begin{cases} q_i^T a_j, & i \ne j \\ \|a_j - \sum_{i=1}^{j-1} r_{ij} q_i\|_2, & i = j \end{cases}$

- Gram Schmidt with projector: $q_n = \frac{P_n a_n}{\|P_n a_n\|}$, where
  $P_n = I - \sum_{i=0}^{n-1} q_i q_i^T = I - \hat{Q}_{j-1}\hat{Q}_{j-1}^T$
- Modified Gram-Schmidt:
  $P_n a_n = (I - \sum_{i=0}^{n-1} q_i q_i^T) a_n = \left[\prod_{i=0}^{n-1} (I - q_i q_i^T)\right] a_n$

Algo: for $j = 1 \to n$:
$v_j^{(1)} = a_j$
$v_j^{(2)} = P_{\perp q_1} a_j = (I - q_1 q_1^T) a_j = v_j^{(1)} - q_1 q_1^T v_j^{(1)}$
$v_j^{(3)} = P_{\perp q_2} a_j = v_j^{(2)} - q_2 q_2^T v_j^{(2)}$
$\vdots$
$v_j^{(j)} = v_j^{(j-1)} - q_{(j-1)} q_{(j-1)}^T v_j^{(j-1)}$
$q_j = \frac{v_j^{(j)}}{\|v_j^{(j)}\|_2}$

- Householder Triangularisation
  - Let $\hat{x} = \begin{bmatrix} - \\ - \\ - \\ \vdots \end{bmatrix}$, F should be s.t. $Fx = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| \cdot e_1$

Hence, $y = Fx = (I - 2uu^T)x$, where $u = -\frac{v}{\|v\|}$,
where $v = \|x\|_2 e_1 - x$

Algo: for $k = 1 \to n$:
$x = A(k:m, k)$ // row k to row m in the k^{th} coln.
$v_k = sgn(x_1) \cdot \|x\|_2 e_1 + x$
$v_k = \frac{v_k}{\|v_k\|}$
$A(k:m, k:n) -= 2 v_k v_k^T A(k:m, k:n)$

Algo	FLOPs	Stability	Error
CGS	$2mn^2$	Unstable	$O(\hat{\kappa}(A)^2 \cdot \varepsilon_m)$
MGS	"	Backward stable	$O(\hat{\kappa}(A) \cdot \varepsilon_m)$
Householder	$2mn^2 - 2/3n^3$	"	$O(\varepsilon_m)$

Linear Least Squares

- If $Ax = b$ is over determined, we can solve for $A^TA\hat{x} = A^Tb$ instead. This shall minimise the residual, i.e. $\|r\|_2$, where $r = A\hat{x} - b$
- $A\hat{x} = Pb$ // get b to coln. space of A
  $A\hat{x} = A(A^TA)^{-1}A^Tb \Rightarrow \hat{x} = (A^TA)^{-1}A^Tb$
- Soln. by Cholesky decomposition:
  $A^TA\hat{x} = A^Tb$
  $R^TR\hat{x} = A^Tb$
  $R^Tw = A^Tb$ // Let $w = R\hat{x}$
  ↓ Solve for w
  $R\hat{x} = w$
  ↓ Solve for $\hat{x}$
  $\hat{x}$
- Soln. by QR factorisation
  $Ax = Pb$
  $QRx = QQ^Tb$
  $Rx = Q^Tb$
  ↓ Solve for x
  x

Algo	Work
Cholesky	$mn^2 + n^3/3$
QR (Householder)	$2mn^2 - \frac{2n^3}{3}$
SVD	$2mn^2 + 11n^3$

- Soln with SVD
  $Ax = Pb$
  $U\Sigma V^Tx = UU^Tb$
  $\Sigma V^Tx = U^Tb$
  $\Sigma y = U^Tb$ // Let $V^Tx = y$
  ↓ Solve for y
  $V^Tx = y$
  ↓ Solve for x
  x
- If A is close to rank-deficient, (Either $m \ge n$ & rank$(A) < n$, or $m < n$)
  $Ax = b$
  $U\Sigma V^Tx = b$ ($U^TU = I$, whereas UU^T will be a projector matrix)
  $\cancel{U^TU}\Sigma V^Tx = U^Tb$ (I)
  $V^Tx = \Sigma^{-1}U^Tb$
  Let $x = V_1y + V_2z$, where $V = [V_1 \; V_2]$ (range(A), Null(A))
  Hence $y = \Sigma^{-1}U^Tb$
  Therefore $x = V_1\Sigma^{-1}U^Tb + V_2z$
  $\|x\|_2 = \|V_1\Sigma^{-1}U^Tb\|_2 + \|V_2z\|_2 - \cancel{\|(V_1\Sigma^{-1}U^Tb, V_2z)\|_2}$ (inner product $= 0$)
  To minimise $\|x\|_2$, we set $z = 0$
- If $b \notin Range(A)$, then we solve $Ax = Pb$.
  In this case, $x = V_1\Sigma^{-1}U^TPb + V_2z$ ($P = UU^T$)
  $= V_1\Sigma^{-1}U^Tb + V_2z$

Similarity Transformation

- If $X \in R^{m \times m}$ is non-singular, XAX^{-1} is known as similarity transformation of A.
- Two matrices A & B are said to be similarity if there exist a similarity transformation bet. them, i.e $B = XAX^{-1}$.
  - A & B will have same eigenvalues & respective multiplicities

$P_B(z) = \det(zI - XAX^{-1})$
$= \det(zXX^{-1} - XAX^{-1})$
$= \det(X(zI - A)X^{-1})$
$= \cancel{\det(X)}\det(zI - A)\cancel{\det(X^{-1})}$ (shown)

* Eigen vectors may not be the same for A & B

Defective Eigenvalues & Matrices

- An eigenvalue, for which algebraic multiplicity > geometric multiplicity is a defective eigenvalue.
- Any matrix that has a defective eigenvalue is a defective matrix
  - It does not posses a full set of L.I. eigenvectors.
- Diagonal matrices are not defective
- Diagonalisability: If $A \in R^{m \times n}$ is not defective iff it has eigenvalue decomposition.
- Unitary diagonalisability (orthogonal diagonalisability): If a non-defective matrix A has eigenvalue decomposition $A = Q\Lambda Q^{-1} = Q\Lambda Q^{\perp}$, where Q is a unitary matrix. (not all non-defective matrices are unitary diagonalisable)
  $Q \in \mathbb{C}^{m \times m}$ $\quad Q^{\perp}Q = QQ^{\perp} = I$ (conjugate)
- Symmetric matrices have all real eigenvalues & eigen vectors.
- Skew symmetric matrices have all imaginary eigen values. Skew symmetric matrices are also unitary diagonalisable.

Symmetric Matrices

- A real symmetric matrix is non-defective & unitary diagonalisable, with real eigenvalues.
- A real skew-symmetric matrix is also non-defective & unitary diagonalisable, with purely complex eigenvalues.
- Any normal matrix $A \in \mathbb{C}^{m \times m}$, s.t. $A^{\perp}A = AA^{\perp}$, will be unitary diagonalisable.

Schur Factorisation

- $A = QTQ^{\perp}$, where Q is unitary, and T is U.T.M.
- SVD of T = SVD of A.
- Every sq. matrix has a Schur factorisation.

* If A is real, A can be decomposed to UTU^T, where U & T are real, and T is quasi U.T.M.

- Schur factorisation need not be unique.

Eigen solvers

- Phase 1: Reduce A to upper Hessenberg matrix H. (UTM but with additional line of non-zero elements parallel to diagonal). $O(m^3)$ flops
- Phase 2: Reduce H to U.T.M. $O(m)$ iterations, $O(m^2)$ flops per iteration $\Rightarrow O(m^3)$ flops.

* Without phase 1, we would need $O(m^4)$ flops.

- Phase 1: Reduce A to H as follows: $A = QHQ^{\perp}$, where $Q = Q_1 \dots Q_{m-2}$.
- Rayleigh quotient: $\alpha = \frac{x^TAx}{x^Tx}$
  - α will be the eigenvalue of A closest to x. This is the least sq. soln. that minimises $\|Ax - \alpha x\|_2$.

Power Iterations

→ Finds the eigenvector corresponding to largest eigenvector (by magnitude).

Algo: Initialise $v^{(0)}$ to a random unit vector

for $k = 1 \to \infty$,

$w = A v^{(k-1)}$

$v^{(k)} = \dfrac{w}{\|w\|}$

$\lambda^{(k)} = (v^{(k)})^T A (v^{(k)})$

Results:

$\| v^{(k)} - (\pm q_1) \|_2 = O\left(\left| \dfrac{\lambda_2}{\lambda_1} \right|^k \right)$

$|\lambda^{(k)} - \lambda_1| = O\left(\left| \dfrac{\lambda_2}{\lambda_1} \right|^{2k} \right)$

If k is even, $v^{(k)} \to q_1$, otherwise, $v^{(k)} \to -q_1$.

* Convergence is slow if $\lambda_2 \simeq \lambda_1$

Rayleigh Quotient Iteration

Algo: Initialise $v^{(0)}$ to some random unit vector

$\lambda^{(0)} = (v^{(0)})^T A (v^{(0)})$

for $k = 1 \to \infty$:

$w = (A - \lambda^{(k-1)} I)^{-1} v^{(k-1)}$ // again linear system of eqns.

$q^{(k)} = \dfrac{w}{\|w\|}$

$\lambda^{(k)} = (q^{(k)})^T A (q^{(k)})$

→ Very fast convergence:

$$\| v^{(k+1)} - (\pm q_j) \| = O\left(\| v^{(k)} - (\pm q_j) \|^3 \right)$$

$$|\lambda^{(k+1)} - \lambda_j| = O\left(|\lambda^{(k)} - \lambda_j|^3 \right)$$

Multiple Eigen values

→ Subspace / simultaneous iterations

→ Take multiple vectors which are L.I. Provided we have an ∞ precision computer, these will converge to different eigenvectors.

→ Assumption #1: The first n eigenvalues are distinct & well-separated

→ " #2: If $\hat{Q}_1 = [q_1 \cdots q_n]$, where $\{q_i \cdots q_n\}$ are eigenvectors of A, $\hat{Q}_1^T v^{(0)}$ is non-singular, and all principal submatrices of $\hat{Q}_1^T v^{(0)}$ are also singular.

→ Orthogonalise at each step to prevent loss of orthogonality, and hence make the algo. stable.

→ Works for large, sparse matrices

Algo: Initialise $\hat{Q}^{(0)} \in \mathbb{R}^{m \times n}$

for $k = 1 \to \infty$:

$Z^{(k)} = A \hat{Q}^{(k-1)}$

$\hat{Q}^{(k)}, \hat{R}^{(k)} = Z$ // QR factorisation

→ Pure QR algorithm (dense matrices)

Algo: $A^{(0)} = A$

for $k = 1 \to \infty$

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

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

→ As $k \to \infty$, $A^{(k)}$ approaches Schur form.

→ Mathematically equivalent to simultaneous iterations

→ Can also be considered a simultaneous inverse iteration applied to a flipped identity matrix.

Inverse Power Iterations

Algo: Initialise μ = some value near λ_J,

$v^{(0)}$ = " random unit vector

for $k = 1 \to \infty$:

$w = (A - \mu I)^{-1} v^{(k-1)}$ // solve by forming soln to system of lin eqns.

$v^{(k)} = \dfrac{w}{\|w\|}$

$\lambda^{(k)} = (v^{(k)})^T A (v^{(k)})$

Converges to closest eigenvalue of μ. If λ_J is the closest to μ and λ_K is second-most closest to μ,

$\| v^{(k)} - (\pm q_J) \| = O\left(\left| \dfrac{\mu - \lambda_J}{\mu - \lambda_K} \right|^k \right)$

$|\lambda^{(k)} - \lambda_J| = O\left(\left| \dfrac{\mu - \lambda_J}{\mu - \lambda_K} \right|^{2k} \right)$

Analysis of Algos. (per iteration)

→ Power iteration: $O(m^2)$ due to matrix-vector multiplication.

→ Inverse power iteration: $O(m^3)$ due to soln. of linear system of eqns.

→ Can be reduced to $O(m^2)$ by solving $(A - \mu I)^{-1}$ once

→ Rayleigh quotient iteration: $O(m^3)$, but lesser iterations are reqd.

→ Can be reduced to $O(m^2)$ by reducing A to tridiagonal / upper Hessenberg.

→ Modified QR (most used by eigensolvers)

Full Algo: Define $A^{(0)}$ s.t. $(Q^{(0)})^T A^{(0)} (Q^{(0)}) = A$ // tridiagonalisation of A

for $k = 1 \to \infty$:

Pick a shift $\mu^{(k)}$ // many methods for picking, e.g. $\mu^{(k)} = A^{(k-1)}_{mm}$

$Q^{(k)} R^{(k)} = A^{(k-1)} - \mu^{(k)} I$ // shifted QR factorisation

$A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$

If any off-diagonal entries are close to 0, set $A_{j,j+1} = A_{j+1,j} = 0$

Split $A^{(k)}$ into A_1 & A_2 s.t. $A^{(k)} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}$

Apply QR algo (from tridiagonalisation) on A_1 & A_2.

→ Krylov subspace method (fully iterative):

→ Krylov subspace is a subspace rich in eigenvectors. This is the set of vectors $\{b, Ab, A^2 b, \ldots\}$.

→ This seq. is similar to power iterations

→ For this to be an actual subspace, $b, Ab, A^2 b$ etc. must be L.I. They are confirmed to be L.I. if A is full-rank.

→ This method is computationally unstable

→ Arnoldi Iteration (To construct Krylov subspace)

Algo: b = arbitrary vector

$q_1 = \dfrac{b}{\|b\|}$

for $n = 1 \to \infty$

$v = A q_1$

for $j = 1 \to n$

$h_{jn} = q_j^T v$

$v = v - h_{jn} q$

$h_{(n+1)n} = \|v\|$

$q_{n+1} = \dfrac{v}{h_{(n+1)n}}$

→ At the end of the iterations, we have:

→ Constructed a subspace rich in eigenvalues of A

→ Projected A onto the subspace, to obtain H_n

→ Hence, H_n is a projection of A onto K_n

→ Eigenvalues of H_n are Arnoldi eigenvalue estimates, a.k.a. Ritz values

→ Arnoldi iterations can be viewed as polynomial approximation

→ Arnoldi approximation problem: Find $p^n \in P^n$ s.t. $\| p^n(A) b \|_2$ is minimum. P^n is the set of monic polynomials of deg. n.

→ Soln. to this problem is actually $P_{H_n}(z) = \det(zI - H_n)$

→ As $n \to \infty$, the soln. approaches eigenvalues of A.