Review of numerical linear algebra material

A. Donev, Spring 2021

To solve linear square system

$$\overleftrightarrow{A}\,\vec{x} = \vec{b}, \quad A \text{ is } \underline{\text{invertible}},$$

$$\sum_{j=1}^{n} a_{ij}\, x_j = b_i \qquad \left[a_{ij}^{(1)} = a_{ij}\right]$$

we use Gaussian elimination, which multiplies each row by

$$\underset{\substack{\uparrow \quad \uparrow \\ \text{eq} \quad \text{var}}}{l_{ik}} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \Leftarrow \text{step } k, \quad i > k$$

and subtracts from another row

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik}\, a_{kj}^{(k)} \qquad i,j > k \tag{1}$$

If we put the ℓ's together we form a lower triangular matrix

$$\begin{cases} L_{ik} = \ell_{ik} & , i > k \\ L_{ii} = 1 \end{cases}$$

and what remains of A is an upper triangular matrix

$$U_{ik} = a^{(i)}_{ik} \quad , i \geq k$$

This gives us the LU factorization of a square invertible matrix

$$A = L\,U$$

L ↑ unit lower triangular

U ← upper triangular, non zeros on diagonal

2

determinant $\rightarrow |A| = \prod_{i=1}^{n} U_{ii} \neq 0$

If we encounter a zero on the diagonal, $a_{kk}^{(k)} = 0$, we do row pivoting (swap order of equations) to find the <u>largest</u> (in magnitude) value below the diagonal element.

$$LU = PA = \text{permuted } A \text{ (rows swapped)}$$

Make sure you can perform pivoted LU factorization of a small or a large but simple matrix

Once we have LU, to solve $Ax = b$ we do

$$L(Ux) = Ly = b$$

$Ly = b$ solve first using forward substitution

Then solve

$Ux = y$ using backward substitution

LU factorization costs $O(n^3)$ (with a prefactor of $O(1)$) FLOPs — expensive for large n but cheaper than any other factorization for a general matrix.

Backward / Forward substitution costs $O(n^2)$ FLOPS = much cheaper

Conditioning number of a matrix

$$K(A) = \|A\| \|A^{-1}\|$$

(in different norms, 1, 2, ∞)

Uncertainty of r.h.s. $b \leftarrow b + \delta b$ causes uncertainty in the answer $x \leftarrow x + \delta x$

$$\frac{\|\delta x\|}{\|x\|} \leq K(A) \frac{\|\delta b\|}{\|b\|}$$

$$\geqslant 10^{-16} \cdot K(A)$$

in double precision

We loose $\log_{10}(K)$ digits of accuracy in x.

However, if we do pivoting,

$$\|Ax - b\| \sim 10^{-16} \|b\|$$

i.e., we find some x that solves $Ax=b$ to roundoff error.

If $A \in \mathbb{R}^{m \times n}$ is not square, what we mean by

$$Ax=b \quad , m > n$$

is least squares solution

$$x = \arg\min_{\tilde{x}} \|A\tilde{x}-b\|_2^2$$

Two ways to solve. First, normal equations

$$\underbrace{(A^TA)}_{\text{invertible is } A \text{ is full-rank}} x = A^Tb$$

$n \times n$ linear system

This is simple but squares the conditioning number & is not the best on a computer.

Cost of normal equations is

$$O(m \cdot n^2) + O(n^3) = O(m \cdot n^2) \text{ FLOPS}$$

since $m > n$

Better approach is to use QR factorization,

$$A = QR$$

Q: orthogonal

R: upper triangular, nonzeros on diagonal

$$Q^T Q = I$$

To solve for least squares x:

$$Rx = Q^T b$$ (backward substitution)

To compute matrix Q, i.e., to find orthonormal basis for column space of A, we use

Gram - Schmidt orthogonalization

$$\tilde{q}_1 = a_1$$

$$\tilde{q}_{k+1} = a_{k+1} - \mathrm{Proj}_{\{q_1, \dots, q_k\}} a_{k+1}$$

$$= a_{k+1} - \sum_{j=1}^{k} (a_{k+1} \cdot q_j) q_j$$

and normalize

$$q_{k+1} = \tilde{q}_{k+1} / \|\tilde{q}_{k+1}\|_2$$

We can do this for any vector space, not just range(A), and any inner product

We use least squares to <u>fit</u> linear models to data

$$y(x) = \sum_{k=1}^{n} a_k f_k(x)$$

unknown coefficients

some linearly independent functions e.g. $\{1, x^1, x^2, \ldots, x^{n-1}\}$

important that unknowns enter <u>linearly</u> = linear least squares

$$y_j = \sum_{k=1}^{n} \left(f_k(x_j) \right) a_k$$

$j = 1, .., m$

$$\vec{y} = \overleftrightarrow{V} \vec{a}$$

$$V_{kj} = f_k(x_j)$$

is "Vandermonde"-like matrix

(<u>not</u> square in general)

Be able to formulate linear least squares fitting problems & solve them for given data

If a square matrix is diagonalizable (non-defective), it has an eigenvalue decomposition

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

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

similarity transformation

diagonal matrix

$\Lambda_{kk} = \lambda_k$ = eigenvalues

$X_{:,k} = x_k$ = eigenvectors as columns

$A x_k = \lambda_k x_k$ (definition)

If A is unitarily diagonalizable, as <u>all</u> symmetric matrices are,

$$A = U \Lambda U^*$$

unitary (orthogonal) matrix

$$U^* U = U U^* = I \Rightarrow$$

$$U^{-1} = U^*$$

columns are orthonormal

$$A^{-1} = (U^*)^{-1} \Lambda^{-1} U^{-1}$$

$$= (U^{-1})^* \Lambda^{-1} U^{-1}$$

$$= (U^*)^* \Lambda^{-1} U^{-1} = U \Lambda^{-1} U^*$$

More generally,

$$f(A) = U \underbrace{f(\Lambda)}_{\substack{\text{apply } f \text{ to} \\ \text{diagonal elements} \\ \text{(eigenvalues)}}} U^*$$

E.g. $\exp(A)$ is used to solve linear systems of ODEs

Every matrix has an SVD decomposition

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

$$[m \times n] = [m \times m]\,[m \times n]\,[n \times n]$$

or reduced SVD (more practical)

$$[m \times n] = [m \times n]\,[n \times n]\,[n \times n] \quad \text{if } m > n$$

$$[m \times n] = [m \times m]\,[m \times m]\,[m \times n] \quad \text{if } n > m$$

U and V are unitary / orthogonal matrices

$$\Sigma = \mathrm{Diag}\{\sigma_1, \sigma_2, \dots, \sigma_{\min\{m,n\}}\}$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$$

$$p = \min(m, n)$$

SVD reveals rank of A:

$$\begin{cases} \sigma_1, \dots, \sigma_r > 0 \\ \sigma_{r+1}, \sigma_{r+2}, \dots = 0 \end{cases}$$

Also allows us to define a matrix pseudoinverse for any matrix

$$A^{\dagger} = V \Sigma^{\dagger} U^*$$

$$\Sigma^{\dagger} = \mathrm{Diag}\{\sigma_1^{-1}, \sigma_2^{-1}, \dots, \sigma_r^{-1}, 0, \dots, 0\}$$

$$X = A^{+} b$$

works for square systems (helps with ill-conditioning), overdetermined and even underdetermined systems.

How do we compute eigenvalue decomposition?

Power method gives us the eigenvalue of largest magnitude & its eigenvector:

Choose <u>random</u> q_0

$$\begin{cases} \tilde{q}_k = A q_{k-1} \\ q_k = \tilde{q}_k / \|\tilde{q}_k\| \approx x_1 \end{cases}$$

$$\lambda_k = \underbrace{q_k^*}_{\text{complex conjugate transpose}} A\, q_k \simeq \lambda_1$$

Converges iff

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots$$

and $q_0 \cdot x_1 \neq 0$

For finding all eigenvalues use computer code, not by hand. (QR algorithm).

To find SVD, do eigenvalue decomposition of normal matrix

$$\begin{cases} A A^* = U |\Sigma|^2 U^* \\ A^* A = V |\Sigma|^2 V^* \end{cases}$$

$$\Rightarrow \sigma_i = \sqrt{\lambda_i(A^*A)}$$

$$= \sqrt{\underset{\substack{\vee\!\!/\!\!/ \\ 0}}{\lambda_i}(AA^*)} \geq 0$$

Matrix norm induced by vector norm

$$\|A\| = \sup_{\|x\|=1} \|Ax\|$$

$$\|Ax\| \leq \|A\| \, \|x\|$$

$$\|A\|_2 = \sigma_1 = \text{largest singular value}$$

$$\Rightarrow K_2(A) = \frac{\sigma_1}{\sigma_n} \quad \text{(conditioning number)}$$