MA580H Matrix Computations

Lectures 1 & 2: Vectors and Matrices

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

Topics:

- Vectors in $\mathbb{R}^n$ and $\mathbb{C}^n$
- Matrix-vector multiplication
- Matrix-matrix multiplication
- Block matrices
- Outer product of vectors

Course Syllabus

Linear systems: All variants of Gaussian elimination and LU factorization, Cholesky factorization.

Linear least-squares problem: Normal equations, rotators and reflectors, QR factorization via rotators, reflectors and Gram Schmidt orthonormalisation, QR method for linear least-squares problems, rank deficient least-squares problems.

Singular value decomposition (SVD): Numerical rank determination via SVD, solution of least squares problems, Moore- Penrose inverse, low rank approximations via SVD, Principal Component Analysis, applications to data mining and image recognition.

Eigenvalue Decomposition: Power, inverse power and Rayleigh quotient iterations, Schur's decomposition, unitary similarity transformation of Hermitian matrices to tridiagonal form, QR algorithm, implementation of explicit QR algorithm for Hermitian matrices.

Textbooks

- L. N. Trefethen and David Bau, Numerical Linear Algebra, SIAM, Philadelphia, 1997.
- D. S. Watkins, Fundamentals of Matrix Computations, 2nd Edition, Wiley, 2002.
- L. Elden, Matrix Methods in Data Mining and Pattern Recognition, SIAM, Philadelphia, 2007.

Another good book on Least-Squares problems:

- S. Boyd and L. Vandenberghe, Introduction to Applied Linear Algebra: Vectors, Matrices and Least Squares, Cambridge University Press, 2018

Vectors in $\mathbb{R}^n$

We define $\mathbb{R}^n$ to be the set of all ordered n-tuples of real numbers. Thus an n-tuple in $\mathbb{R}^n$ (also called an n-vector) is of the form

$$\text{row vector:} \quad \mathbf{v} = [v_1, \ldots, v_n] \quad \text{or} \quad \text{column vector:} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

We always write a vector in $\mathbb{R}^n$ as a column vector. Thus

$$\mathbb{R}^n := \left\{ \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} : v_1, \ldots, v_n \in \mathbb{R} \right\}.$$

$$\text{Transpose:} \quad [v_1, \ldots, v_n]^\top = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \text{ and } \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}^\top = [v_1, \ldots, v_n].$$

Vectors in $\mathbb{C}^n$

We define $\mathbb{C}^n$ to be the set of all ordered n-tuples of complex numbers. Thus an n-tuple in $\mathbb{C}^n$ (also called an n-vector) is of the form

$$\text{row vector:} \quad \mathbf{v} = [v_1, \ldots, v_n] \quad \text{or} \quad \text{column vector:} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$$

We always write a vector in $\mathbb{C}^n$ as column vector. Thus

$$\mathbb{C}^n := \left\{ \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} : v_1, \ldots, v_n \in \mathbb{C} \right\}$$

Conjugate transpose: Here $\bar{z}$ is the complex conjugate of $z \in \mathbb{C}$.

$$[v_1, \ldots, v_n]^* = \begin{bmatrix} \bar{v}_1 \\ \vdots \\ \bar{v}_n \end{bmatrix} \text{ and } \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}^* = [\bar{v}_1, \ldots, \bar{v}_n].$$

Algebraic properties of vectors in $\mathbb{R}^n$ and $\mathbb{C}^n$

Define addition and scalar multiplication on $\mathbb{F}^n$ ($\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$) as follows:

$$\begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} + \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ \vdots \\ u_n + v_n \end{bmatrix} \text{ and } \alpha \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} \alpha u_1 \\ \vdots \\ \alpha u_n \end{bmatrix} \text{ for } \alpha \in \mathbb{F}.$$

This produces new vectors from old vectors. For $\mathbf{u}, \mathbf{v}, \mathbf{w}$ in $\mathbb{F}^n$ and scalars α, β in $\mathbb{F}$, the following hold:

1. Commutativity: $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
2. Associativity: $(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$
3. Identity: $\mathbf{u} + \mathbf{0} = \mathbf{u}$
4. Inverse: $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$
5. Distributivity : $\alpha(\mathbf{u} + \mathbf{v}) = \alpha\mathbf{u} + \alpha\mathbf{v}$
6. Distributivity : $(\alpha + \beta)\mathbf{u} = \alpha\mathbf{u} + \beta\mathbf{u}$
7. Associativity: $\alpha(\beta\mathbf{u}) = (\alpha\beta)\mathbf{u}$
8. Identity: $1\mathbf{u} = \mathbf{u}$.

Examples of vectors

Standard vectors: The vectors $\mathbf{e}_1 := \begin{bmatrix} 1 & 0 & \cdots 0 \end{bmatrix}^\top$, $\mathbf{e}_2 := \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^\top, \ldots, \mathbf{e}_n := \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^\top$ are called standard vectors or canonical vectors in $\mathbb{R}^n$ and $\mathbb{C}^n$.

Features vectors. A feature vector collects together n different quantities that pertain to a single thing or object. The entries of a feature vector are called the features or attributes.

For instance, a 5-vector $\mathbf{x} := [x_1, x_2, x_3, x_4, x_5]^\top$ could give the age, height, weight, blood pressure, and temperature of a patient admitted to a hospital.

Word count vector. An n-vector $\mathbf{w}$ can represent the number of times each word in a dictionary of n words appears in a document.

For instance, the word count vector $[25, 2, 0]^\top$ means that the first dictionary word appears 25 times, the second one twice, and the third one not at all.

Matrices

Definition: A matrix is an array of numbers. An $m \times n$ matrix A has m rows and n columns and is of the form

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

The j-th column of A : $\mathbf{a}_j := \begin{bmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{bmatrix}$ for $j = 1 : n$.

The i-th row of A: $\hat{\mathbf{a}}_i := \begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{bmatrix}$ for $i = 1 : m$. Then

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = \left[\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array} \right] = \begin{bmatrix} - \hat{\mathbf{a}}_1 - \\ \vdots \\ - \hat{\mathbf{a}}_m - \end{bmatrix}.$$

Special matrices

An $m \times n$ matrix said to be a square matrix if $m = n$. An $m \times n$ matrix $D := [d_{ij}]$ is said to be a diagonal matrix if $d_{ij} = 0$ for all $i \neq j$. An $n \times n$ diagonal matrix D with diagonal entries $d_1, \ldots, d_n$ is given by

$$D = \operatorname{diag}(d_1, \ldots, d_n) = \begin{bmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{bmatrix}.$$

Identity matrix: An $n \times n$ diagonal matrix with all diagonal entries equal to 1 is called the identity matrix and is denoted by I_n or I.

Zero matrix: An $m \times n$ matrix with all entries 0 is called the zero matrix and is denoted by $\mathbf{O}_{m \times n}$ or simply by $\mathbf{O}$.

Example: $I := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{O} := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$.

Matrix addition and scalar multiplication

Let $\mathbb{F}^{m\times n}$ denote the set of all $m \times n$ matrices with entries in $\mathbb{F}$ where $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$. Let $A := [a_{ij}]$ and $B := [b_{ij}]$ be matrices $\in \mathbb{F}^{m\times n}$ and $\alpha \in \mathbb{F}$.

1. Matrix addition: $A + B := [a_{ij} + b_{ij}] \in \mathbb{F}^{m\times n}$.
2. Multiplication by a scalar: $\alpha\, A := [\alpha\, a_{ij}] \in \mathbb{F}^{m\times n}$.

Let $A := \begin{bmatrix} 1 & 4 & 0 \\ -2 & 6 & 5 \end{bmatrix}$ and $B := \begin{bmatrix} -3 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix}$. Then

$$A+B = \begin{bmatrix} 1 & 4 & 0 \\ -2 & 6 & 5 \end{bmatrix} + \begin{bmatrix} -3 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} -2 & 5 & -1 \\ -2 & 6 & 7 \end{bmatrix}$$

$$2A = \begin{bmatrix} 2 & 8 & 0 \\ -4 & 12 & 10 \end{bmatrix} \text{ and } (-1)A = \begin{bmatrix} -1 & -4 & 0 \\ 2 & -6 & -5 \end{bmatrix}.$$

Transpose and Conjugate transpose

Transpose: The transpose of an $m \times n$ matrix $A = [a_{ij}]_{m\times n}$ is the $n \times m$ matrix denoted by $A^\top$ and is given by $A^\top = [a_{ji}]_{n\times m}$.

Example: $\begin{bmatrix} 1 & 2 \\ 4 & 5 \\ 7 & 8 \end{bmatrix}^\top = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \end{bmatrix}$ and $\begin{bmatrix} 1+i & 2 \\ 3 & 4+5i \end{bmatrix}^\top = \begin{bmatrix} 1+i & 3 \\ 2 & 4+5i \end{bmatrix}$

Conjugate transpose: The conjugate transpose of an $m \times n$ complex matrix $A = [a_{ij}]_{m\times n}$ is the $n \times m$ matrix denoted by A^* and is given by

$$A^* = [\bar{a}_{ji}]_{n\times m} = ([\bar{a}_{ij}]_{m\times n})^\top = (\bar{A})^\top,$$

where $\bar{a}_{ij}$ is the complex conjugate of a_{ij}.

Example: $\begin{bmatrix} i & 4 & 1+i \\ 3 & 4+5i & 0 \end{bmatrix}^* = \begin{bmatrix} -i & 3 \\ 4 & 4-5i \\ 1-i & 0 \end{bmatrix}$

Transpose and conjugate transpose

Exercise: Let $A, B \in \mathbb{F}^{m\times n}$ and $\alpha \in \mathbb{F}$. Then show that

$$(a)\,(A+B)^\top = A^\top + B^\top \quad (b)\,(\alpha A)^\top = \alpha A^\top \text{ and } (\alpha A)^* = \bar{\alpha} A^* \quad (c)\,(A^\top)^\top = A.$$

Definition: Let A be an $n \times n$ matrix. Then A is said to be

1. symmetric if $A^\top = A$
2. skew-symmetric if $A^\top = -A$
3. Hermitian if $A^* = A$
4. skew-Hermitian if $A^* = -A$.

Remark: Let $A := [a_{ij}]_{n\times n}$. If $A^\top = -A$ then $a_{jj} = 0$ for $j = 1 : n$. On the other hand, if $A^* = -A$ then $\mathrm{Re}(a_{jj}) = 0$ for $j = 1 : n$.

Matrix-vector multiplication

Let $A := \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_n \end{bmatrix} \in \mathbb{F}^{m \times n}$ and $\mathbf{x} := [x_1, \ldots, x_n]^\top \in \mathbb{F}^n$. We define the matrix-vector multiplication $A\mathbf{x}$ to be the linear combination of columns of A.

Definition: Matrix-vector multiplication

$$A\mathbf{x} = \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n.$$

Example:

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ -x_1 + x_2 \\ -x_2 + x_3 \end{bmatrix}.$$

Matrix-vector multiplication

A row vector $\begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix}$ is a $1 \times n$ matrix. Therefore

$$\begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = a_{i1}x_1 + \cdots + a_{in}x_n.$$

Example: Matrix-vector multiplication in two ways

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} x_1 \\ x_1 + x_2 \\ x_2 + x_3 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x} \\ \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \mathbf{x} \\ \begin{bmatrix} 0 & 1 & 1 \end{bmatrix} \mathbf{x} \end{bmatrix}$$

Row and column oriented matrix-vector multiplication

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ldots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} a_{11} \\ \vdots \\ a_{m1} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \end{bmatrix} \mathbf{x} \\ \vdots \\ \begin{bmatrix} a_{m1} & \cdots & a_{mn} \end{bmatrix} \mathbf{x} \end{bmatrix}.$$

Writing $A := \begin{bmatrix} \mathbf{a}_1 \mid \cdots \mid \mathbf{a}_n \end{bmatrix}$ and $A = \begin{bmatrix} -\hat{\mathbf{a}}_1 - \\ \vdots \\ -\hat{\mathbf{a}}_m - \end{bmatrix}$, we have

$$A\mathbf{x} = x_1\mathbf{a}_1 + \cdots + x_n\mathbf{a}_n = \begin{bmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}_1\mathbf{x} \\ \vdots \\ \hat{\mathbf{a}}_m\mathbf{x} \end{bmatrix}.$$

Matrix-matrix multiplication

Fact: Let $A \in \mathbb{F}^{m \times n}$. Let $\mathbf{e}_i \in \mathbb{F}^m$ and $\mathbf{e}_j \in \mathbb{F}^n$ be standard unit vectors. Then

- $A\mathbf{e}_j$ is the j-th column of A.
- $\mathbf{e}_i^\top A$ is the i-th row of A.

Let $A \in \mathbb{F}^{m \times n}$ and $B := \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_p \end{bmatrix} \in \mathbb{F}^{n \times p}$.

Definition: Define the matrix-matrix multiplication AB by

$$AB := \begin{bmatrix} A\mathbf{b}_1 & \cdots & A\mathbf{b}_p \end{bmatrix}.$$

Reason: Define AB to be the $m \times p$ matrix such that $(AB)\mathbf{x} = A(B\mathbf{x})$ for all $\mathbf{x} \in \mathbb{F}^p$. Let $C := AB$ be given by $C = \begin{bmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_p \end{bmatrix}$. Let $\mathbf{e}_j \in \mathbb{F}^p$ be the standard unit vector. Then for $j = 1 : p$, we have $B\mathbf{e}_j = \mathbf{b}_j$ and

$$\mathbf{c}_j = C\mathbf{e}_j = (AB)\mathbf{e}_j = A(B\mathbf{e}_j) = A\mathbf{b}_j \Longrightarrow C = \begin{bmatrix} A\mathbf{b}_1 & \cdots & A\mathbf{b}_p \end{bmatrix}.$$

Matrix-matrix multiplication

Let $A = \begin{bmatrix} -\hat{\mathbf{a}}_1- \\ \vdots \\ -\hat{\mathbf{a}}_m- \end{bmatrix} \in \mathbb{F}^{m\times n}$, $B := \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_p \end{bmatrix} \in \mathbb{F}^{n\times p}$. Then

$$AB = \begin{bmatrix} A\mathbf{b}_1 & \cdots & A\mathbf{b}_p \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}_1\mathbf{b}_1 & \cdots & \hat{\mathbf{a}}_1\mathbf{b}_p \\ \vdots & \cdots & \vdots \\ \hat{\mathbf{a}}_m\mathbf{b}_1 & \cdots & \hat{\mathbf{a}}_m\mathbf{b}_p \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{a}}_1 B \\ \vdots \\ \hat{\mathbf{a}}_m B \end{bmatrix}.$$

Thus if $A := [a_{ij}]_{m\times n}$, $B := [b_{ij}]_{n\times p}$ and $C := AB = [c_{ij}]_{m\times p}$ then

$$c_{ij} = \hat{\mathbf{a}}_i\mathbf{b}_j = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} = \sum_{k=1}^{n} a_{ik}b_{kj}.$$

Remark: If A and B are $n \times n$ matrices then in general $AB \neq BA$.

Example

Let $A = \begin{bmatrix} 1 & 3 & 2 \\ 0 & -1 & 1 \end{bmatrix}$ and $B := \begin{bmatrix} 4 & -1 \\ 1 & 2 \\ 3 & 0 \end{bmatrix}$. Then

$$A\mathbf{b}_1 = \begin{bmatrix} 1 & 3 & 2 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 13 \\ 2 \end{bmatrix} \text{ and } A\mathbf{b}_2 = \begin{bmatrix} 1 & 3 & 2 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \end{bmatrix}.$$

Therefore $AB = \begin{bmatrix} A\mathbf{b}_1 & A\mathbf{b}_2 \end{bmatrix} = \begin{bmatrix} 13 & 5 \\ 2 & -2 \end{bmatrix}$. On the other hand

$$\widehat{\mathbf{a}}_1 B = \begin{bmatrix} 1 & 3 & 2 \end{bmatrix} \begin{bmatrix} 4 & -1 \\ 1 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 13 & 5 \end{bmatrix} \text{ and } \widehat{\mathbf{a}}_2 B = \begin{bmatrix} 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 4 & -1 \\ 1 & 2 \\ 3 & 0 \end{bmatrix} = \begin{bmatrix} 2 & -2 \end{bmatrix}.$$

Therefore $AB = \begin{bmatrix} \widehat{\mathbf{a}}_1 B \\ \widehat{\mathbf{a}}_2 B \end{bmatrix} = \begin{bmatrix} 13 & 5 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} A\mathbf{b}_1 & A\mathbf{b}_2 \end{bmatrix}$.

Properties of matrix multiplication

Thoerm: Let A, B and C be matrices (whose sizes are such that the indicated operations can be performed) and let α be a scalar. Then

1. Associative Law: $(AB)C = A(BC)$
2. Left Distributive Law: $A(B + C) = AB + AC$
3. Right Distributive Law: $(A + B)C = AC + BC$
4. Scalar multiplication: $\alpha(AB) = (\alpha A)B = A(\alpha B)$
5. Multiplicative identity: If A is an $m \times n$ matrix then $I_m A = A = A I_n$.

Block matrices

Definition: An $m \times n$ block matrix (or a partitioned matrix) is a matrix of the form

$$A := \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \cdots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$$

where each A_{ij} is a $p_i \times q_j$ matrix for $i = 1 : m$ and $j = 1 : n$.

Then $\begin{bmatrix} A_{i1} & \cdots & A_{in} \end{bmatrix}$ is the i-th block row of A and $\begin{bmatrix} A_{1j} \\ \vdots \\ A_{mj} \end{bmatrix}$ is the j-th block column of A.

Example: $\left[\begin{array}{cc|ccc|c} 1 & 2 & 2 & 0 & 1 & 4 \\ 3 & 4 & 1 & 2 & 3 & 5 \\ \hline 5 & 7 & 2 & 7 & 8 & 8 \\ 3 & 4 & 1 & 9 & 2 & 2 \end{array}\right]$ has 2 block rows and 3 block columns.

Block matrix operations

Block matrix addition: Let $A := [A_{ij}]_{m\times n}$ and $B := [B_{ij}]_{m\times n}$ be block matrices such that size of A_{ij} = size of B_{ij} for $i = 1 : m$ and $j = 1 : n$. Then $A + B := [A_{ij} + B_{ij}]_{m\times n}$.

Block matrix multiplication: Let $A := [A_{ij}]_{m\times n}$ and $B := [B_{ij}]_{n\times p}$ be block matrices. If the matrix multiplication $C_{ij} := \sum_{k=1}^{n} A_{ik}B_{kj}$ is well defined for $i = 1 : m$ and $j = 1 : p$ then AB is an $m \times p$ block matrix given by $AB = [C_{ij}]_{m\times p}$.

Conformal partition: If an operation on block matrices A and B are well defined then A and B are said to be partitioned conformably.

Example: $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix} =$

$$\begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & A_{11}B_{12} + A_{12}B_{22} & A_{11}B_{13} + A_{12}B_{23} \\ A_{21}B_{11} + A_{22}B_{21} & A_{21}B_{12} + A_{22}B_{22} & A_{21}B_{13} + A_{22}B_{23} \end{bmatrix}.$$

Block matrix multiplication

Example:

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ \hline 3 & 3 & 2 & 2 \end{array}\right] \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ \hline 3 & 1 & 1 & 1 \\ 3 & 2 & 1 & 2 \end{array}\right] = \left[\begin{array}{cc|cc} 8 & 6 & 4 & 5 \\ \hline 10 & 9 & 6 & 7 \\ 18 & 15 & 10 & 12 \end{array}\right]$$

$$\left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1 \\ \hline 3 & 3 & 2 & 2 \end{array}\right] \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ \hline 3 & 1 & 1 & 1 \\ 3 & 2 & 1 & 2 \end{array}\right] = \left[\begin{array}{cc|cc} 8 & 6 & 4 & 5 \\ 10 & 9 & 6 & 7 \\ \hline 18 & 15 & 10 & 12 \end{array}\right]$$

Outer product

Given two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^n$, the standard inner product of $\mathbf{x}$ and $\mathbf{y}$ is given by

$$\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + \cdots + x_n y_n = \mathbf{y}^\top \mathbf{x}.$$

Outer product: The matrix product $\mathbf{x}\mathbf{y}^\top$ is an $n \times n$ matrix and is given by

$$\mathbf{x}\mathbf{y}^\top = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & \cdots & x_1 y_n \\ x_2 y_1 & x_2 y_2 & \cdots & x_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n y_1 & x_n y_2 & \cdots & x_n y_n \end{bmatrix}.$$

The product $\mathbf{x}\mathbf{y}^\top$ is called the outer product of $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$.

Outer product

Example: If $\mathbf{x} := \begin{bmatrix} 4 & 1 & 3 \end{bmatrix}^\top$ and $\mathbf{y} := \begin{bmatrix} 3 & 5 & 2 \end{bmatrix}^\top$ then

$$\mathbf{x}\mathbf{y}^\top = \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix} \begin{bmatrix} 3 & 5 & 2 \end{bmatrix} = \begin{bmatrix} 12 & 20 & 8 \\ 3 & 5 & 2 \\ 9 & 15 & 6 \end{bmatrix}.$$

Outer product of matrices:

Let $X := \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{m \times n}$ and $Y := \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_n \end{bmatrix} \in \mathbb{R}^{p \times n}$. Then $XY^\top \in \mathbb{R}^{m \times p}$ can be written as sum of outer products of vectors

$$XY^\top = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} \begin{bmatrix} \mathbf{y}_1^\top \\ \mathbf{y}_2^\top \\ \vdots \\ \mathbf{y}_n^\top \end{bmatrix} = \mathbf{x}_1\mathbf{y}_1^\top + \mathbf{x}_2\mathbf{y}_2^\top + \cdots + \mathbf{x}_n\mathbf{y}_n^\top.$$

Floating-Point Operation (FLOP) count

Vector-vector operations: Let $\alpha \in \mathbb{R}$. Let $\mathbf{x} := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top \in \mathbb{R}^n$ and $\mathbf{y} := \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^\top \in \mathbb{R}^n$. We ignore the lower order terms for flop count.

- $\mathbf{z} \leftarrow \mathbf{x} + \mathbf{y}$ and $\mathbf{d} \leftarrow \alpha \cdot \mathbf{x}$ require n flops
- $\mathbf{z} \leftarrow \alpha \cdot \mathbf{x} + \mathbf{y}$ and $s \leftarrow \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{j=1}^{n} x_j y_j$ require $2n$ flops

Matrix-vector operations: Let $A := \begin{bmatrix} \mathbf{a}_1 & \cdots & \mathbf{a}_n \end{bmatrix} \in \mathbb{R}^{n \times n}$ and $\beta \in \mathbb{R}$.

- $\mathbf{z} \leftarrow A\mathbf{x} = x_1 \mathbf{a}_1 + \cdots + x_n \mathbf{a}_n$ and $\mathbf{d} \leftarrow \alpha \cdot A\mathbf{x} + \beta \cdot \mathbf{y}$ require $2n^2$ flops
- $\mathbf{z} \leftarrow A^\top \mathbf{x} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{x} & \cdots & \mathbf{a}_n^\top \mathbf{x} \end{bmatrix}^\top$ and $\mathbf{d} \leftarrow \alpha \cdot A^\top \mathbf{x} + \beta \cdot \mathbf{y}$ require $2n^2$ flops

Matrix-matrix operations: Let $B := \begin{bmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{bmatrix} \in \mathbb{R}^{n \times n}$ and $C \in \mathbb{R}^{n \times n}$.

- $D \leftarrow AB = \begin{bmatrix} A\mathbf{b}_1 & \cdots & A\mathbf{b}_n \end{bmatrix}$ and $D \leftarrow \alpha \cdot AB + \beta \cdot C$ require $2n^3$ flops
- $D \leftarrow A^\top B$ or $D \leftarrow AB^\top$ and $D \leftarrow \alpha \cdot A^\top B + \beta \cdot C$ require $2n^3$ flops

MA580H Matrix Computations

Lectures 3 & 4: Orthogonal vectors and matrices

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Orthogonal vectors and orthogonal subspaces
- Orthogonal matrices
- Orthogonal decomposition theorem

Inner product

Angle between two n-vectors can be described by using inner product (dot product).

Definition: If $\mathbf{u} := [u_1, \ldots, u_n]^\top$ and $\mathbf{v} := [v_1, \ldots, v_n]^\top$ are n-vectors then the inner product $\langle \mathbf{u}, \mathbf{v} \rangle$ is defined by

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle &:= u_1 v_1 + u_2 v_2 + \cdots + u_n v_n = \mathbf{v}^\top \mathbf{u} \quad \text{when } \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \\ \langle \mathbf{u}, \mathbf{v} \rangle &:= u_1 \bar{v}_1 + u_2 \bar{v}_2 + \cdots + u_n \bar{v}_n = \mathbf{v}^* \mathbf{u} \quad \text{when } \mathbf{u}, \mathbf{v} \in \mathbb{C}^n. \end{aligned}$$

The inner product $\langle \mathbf{u}, \mathbf{v} \rangle$ is also called dot product and is written as $\mathbf{u} \bullet \mathbf{v}$.

Example: If $\mathbf{u} := [1, 2, -3]^\top$ and $\mathbf{v} := [-3, 5, 2]^\top$ then

$$\langle \mathbf{u}, \mathbf{v} \rangle = 1 \cdot (-3) + 2 \cdot 5 + (-3) \cdot 2 = 1.$$

Inner product

Weights, features, and score. Let $\mathbf{f} := \begin{bmatrix} f_1 & \cdots & f_n \end{bmatrix} \in \mathbb{R}^n$ be a feature vector of an object and $\mathbf{w} := \begin{bmatrix} w_1 & \cdots & w_n \end{bmatrix} \in \mathbb{R}^n$ be a weight vector. Then the inner product

$$\langle \mathbf{f}, \mathbf{w} \rangle = w_1 f_1 + \cdots + w_n f_n$$

is the sum of the feature values, scaled by the weights, and is called a score.

Examples:

- Credit score: Let f be a feature vector associated with a loan applicant (e.g., age, income, . . .). Then we might interpret $\langle \mathbf{f}, \mathbf{w} \rangle$ as a credit score, where w_i is the weight given to feature f_i in forming the score.
- Co-occurrence. Let $\mathbf{x}$ and $\mathbf{y}$ be Boolean n-vectors (each entry is either 0 or 1) that describe occurrence. Then the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ gives the total number of co-occurrences.
  For $\mathbf{x} := [0, 1, 1, 1, 1, 1, 1]^\top$ and $\mathbf{y} := [1, 0, 1, 0, 1, 0, 0]^\top$, we have $\langle \mathbf{x}, \mathbf{y} \rangle = 2$, which is the number of common occurrences.

Properties of inner product

Theorem: Let $\mathbf{u}, \mathbf{v}$, and $\mathbf{w}$ be vectors in $\mathbb{C}^n$ and let $\alpha \in \mathbb{C}$. Then

1. $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$ and $\langle \mathbf{u}, \mathbf{u} \rangle = 0 \iff \mathbf{u} = \mathbf{0}$.
2. $\langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}$
3. $\langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle$
4. $\langle \alpha \mathbf{u}, \mathbf{v} \rangle = \alpha \langle \mathbf{u}, \mathbf{v} \rangle$.

Definition: The norm (or length) of a vector $\mathbf{v} := [v_1, \ldots, v_n]^\top$ in $\mathbb{C}^n$ is a nonnegative number $\|\mathbf{v}\|$ defined by

$$\|\mathbf{v}\| := \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} = \sqrt{|v_1|^2 + \cdots + |v_n|^2}.$$

Theorem (Cauchy-Schwarz Inequality): Let $\mathbf{u}$ and $\mathbf{v}$ be n-vectors. Then

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{u}\| \, \|\mathbf{v}\|.$$

Proof for $\mathbb{R}^n$: $p(t) := \|\mathbf{u} + t\mathbf{v}\|^2 = \|\mathbf{u}\|^2 + 2t\langle \mathbf{u}, \mathbf{v} \rangle + \|\mathbf{v}\|^2 t^2 \geq 0$ for all $t \in \mathbb{R}$. Hence discriminant of $p(t)$ is non-positive which yields the result. ■

Unit vectors

Definition: A vector $\mathbf{v}$ in $\mathbb{C}^n$ or $\mathbb{R}^n$ is called a unit vector if $\|\mathbf{v}\| = 1$. If $\mathbf{u}$ is a nonzero vector then $\mathbf{v} := \frac{1}{\|\mathbf{u}\|}\mathbf{u}$ is a unit vector in the direction of $\mathbf{u}$. Indeed,

$$\|\mathbf{v}\| = \|(1/\|\mathbf{u}\|)\mathbf{u}\| = \frac{1}{\|\mathbf{u}\|}\|\mathbf{u}\| = 1.$$

The vector $\mathbf{v}$ is referred to as a normalization of $\mathbf{u}$.

Example: Let $\mathbf{u} := \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$. Then $\|\mathbf{u}\| = \sqrt{4+1+9} = \sqrt{14}$ and

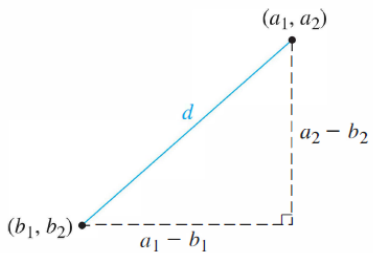
$$\mathbf{v} := \frac{1}{\sqrt{14}}\mathbf{u} = \frac{1}{\sqrt{14}}\begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2/\sqrt{14} \\ -1/\sqrt{14} \\ 3/\sqrt{14} \end{bmatrix}.$$

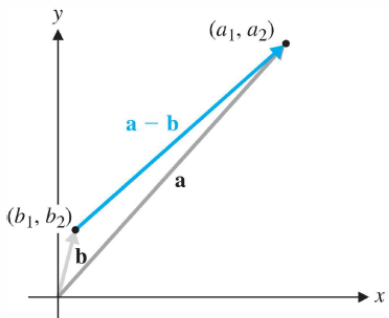
Standard unit vectors: The vectors $\mathbf{e}_1 := [1,0,0]^\top, \mathbf{e}_2 := [0,1,0]^\top$ and $\mathbf{e}_3 := [0,0,1]^\top$ are unit vectors in $\mathbb{R}^3$ and are called standard unit vectors. The canonical vectors $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ in $\mathbb{R}^n$ are standard unit vectors.

Distance

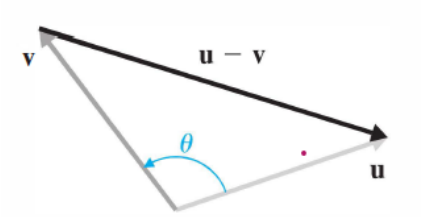
Distance: The distance $\mathrm{d}(\mathbf{u}, \mathbf{v})$ between two vectors $\mathbf{u} := [u_1, \ldots, u_n]^\top$ and $\mathbf{v} := [v_1, \ldots, v_n]^\top$ in $\mathbb{R}^n$ or $\mathbb{C}^n$ is defined by

$$\mathrm{d}(\mathbf{u}, \mathbf{v}) := \|\mathbf{u} - \mathbf{v}\| = \sqrt{|u_1 - v_1|^2 + \cdots + |u_n - v_n|^2}.$$





Angle between two vectors in $\mathbb{R}^2$



Consider the triangle in $\mathbb{R}^2$ with sides $\mathbf{u}, \mathbf{v}$ and $\mathbf{u}-\mathbf{v}$. Let θ be the angle between $\mathbf{u}$ and $\mathbf{v}$. Then by the law of cosines there is unique $\theta \in [0, \pi]$ such that

$$\|\mathbf{u}-\mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\|\mathbf{u}\|\,\|\mathbf{v}\|\cos\theta.$$

Expanding $\|\mathbf{u}-\mathbf{v}\|^2 = \|\mathbf{u}\|^2 - 2\langle \mathbf{u}, \mathbf{v}\rangle + \|\mathbf{v}\|^2$ gives us

$$\langle \mathbf{u}, \mathbf{v}\rangle = \|\mathbf{u}\|\,\|\mathbf{v}\|\cos\theta \Longrightarrow \cos\theta = \frac{\langle \mathbf{u}, \mathbf{v}\rangle}{\|\mathbf{u}\|\,\|\mathbf{v}\|}.$$

Angle between two n-vectors

Definition: Let $\mathbf{u}$ and $\mathbf{v}$ be nonzero n-vectors. Then

$$\cos\theta := \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \Longrightarrow \langle \mathbf{u}, \mathbf{v} \rangle = \|\mathbf{u}\| \, \|\mathbf{v}\| \cos\theta \quad \text{for } \theta \in [0, \pi] \text{ when } \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$$

$$\cos\theta := \frac{|\langle \mathbf{u}, \mathbf{v} \rangle|}{\|\mathbf{u}\| \, \|\mathbf{v}\|} \Longrightarrow |\langle \mathbf{u}, \mathbf{v} \rangle| = \|\mathbf{u}\| \, \|\mathbf{v}\| \cos\theta \quad \text{for } \theta \in [0, \frac{\pi}{2}] \text{ when } \mathbf{u}, \mathbf{v} \in \mathbb{C}^n.$$

Example: Let $\mathbf{u} := \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ and $\mathbf{v} := \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$. Then $\langle \mathbf{u}, \mathbf{v} \rangle = 1 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 1$. We have $\|\mathbf{u}\| = \sqrt{1+1} = \sqrt{2}$ and $\|\mathbf{v}\| = \sqrt{1+1} = \sqrt{2}$. Hence

$$\cos\theta = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \, \|\mathbf{v}\|} = \frac{1}{\sqrt{2}\sqrt{2}} = \frac{1}{2} \Longrightarrow \theta = \pi/3 \text{ radians.} \blacksquare$$

Orthogonal vectors

Definition: Two n-vectors $\mathbf{u}$ and $\mathbf{v}$ are said to be mutually orthogonal if $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. We write $\mathbf{u} \perp \mathbf{v}$ when $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. If, in addition, $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$ then $\mathbf{u}$ and $\mathbf{v}$ are called orthonormal.

Remark: The zero vector $\mathbf{0}$ is orthogonal to all vectors in $\mathbb{R}^n$ as $\langle \mathbf{0}, \mathbf{v} \rangle = 0$ for all $\mathbf{v} \in \mathbb{R}^n$.

Example: The vectors $\mathbf{u} := [1, 1, -2]^\top$ and $\mathbf{v} := [3, 1, 2]^\top$ in $\mathbb{R}^3$ are orthogonal as $\langle \mathbf{u}, \mathbf{v} \rangle = 1 \cdot 3 + 1 \cdot 1 + (-2) \cdot 2 = 0$.

Pythagoras' Theorem: Let $\mathbf{u}$ and $\mathbf{v}$ be n-vectors. Then

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle &= 0 \iff \|\mathbf{u} + \mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 \text{ when } \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \\ \langle \mathbf{u}, \mathbf{v} \rangle &= 0 \implies \|\mathbf{u} + \mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 \text{ when } \mathbf{u}, \mathbf{v} \in \mathbb{C}^n \end{aligned}$$

Proof: We have $\|\mathbf{u} + \mathbf{v}\|^2 = \|\mathbf{u}\|^2 + 2\langle \mathbf{u}, \mathbf{v} \rangle + \|\mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 \iff \langle \mathbf{u}, \mathbf{v} \rangle = 0$ when $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. We have $\|\mathbf{u} + \mathbf{v}\|^2 = \|\mathbf{u}\|^2 + 2\,\mathrm{Re}(\langle \mathbf{u}, \mathbf{v} \rangle) + \|\mathbf{v}\|^2$ when $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$. Hence $\langle \mathbf{u}, \mathbf{v} \rangle = 0 \implies \|\mathbf{u} + \mathbf{v}\|^2 = \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2$. ■

Vectors in information retrieval

Problem: Given a few key words, retrieve relevant information from a large database.

Document vectors: Document vectors are used in information retrieval. Consider the five documents.

Doc. 1:	The Google matrix G is a model of the Internet.
Doc. 2:	G_{ij} is nonzero if there is a link from web page j to i.
Doc. 3:	The Google matrix G is used to rank all web pages.
Doc. 4:	The ranking is done by solving a matrix eigenvalue problem.
Doc. 5:	England dropped out of the top 10 in the FIFA ranking.

The blue colored texts are the key words or terms. The set of terms is called a Dictionary. Counting the frequency of terms in each document, we obtain document vectors.

Term-document matrix

Term	Doc. 1	Doc. 2	Doc. 3	Doc. 4	Doc. 5
eigenvalue	0	0	0	1	0
England	0	0	0	0	1
FIFA	0	0	0	0	1
Google	1	0	1	0	0
Internet	1	0	0	0	0
link	0	1	0	0	0
matrix	1	0	1	1	0
page	0	1	1	0	0
rank	0	0	1	1	1
web	0	1	1	0	1

Each document is a vector in $\mathbb{R}^{10}$ and is represented by a column of the term-document matrix.

Query vector

Suppose that we want to find all documents that are relevant to the query ranking of web pages. This is represented by a query vector, constructed in the way as the document vectors, using the same dictionary:

$$\mathbf{v} := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^\top \in \mathbb{R}^{10}.$$

Thus the query itself is a document. The information retrieval task can now be formulated as a mathematical problem.

Problem: Find the document vectors (columns of the term of document matrix) that are close (in some sense) to the query vector $\mathbf{v}$.

Query matching (use of dot product)

Query matching is the process of finding all documents that are relevant to a particular query $\mathbf{v}$. The cosine of angle between two vectors is often used to determine relevant documents:

$$\cos\theta_j := \frac{|\langle \mathbf{d}_j, \mathbf{v}\rangle|}{\|\mathbf{v}\|\,\|\mathbf{d}_j\|} > \mathrm{tol}$$

where $\mathbf{d}_j$ is the j-th document vector (j-th column of the term-document matrix) and tol is a predefined tolerance. Thus $\cos\theta_j > \mathrm{tol} \Rightarrow \mathbf{d}_j$ is relevant.

For the document vectors $\mathbf{d}_1, \ldots, \mathbf{d}_5$ and the query ("ranking of web pages") vector $\mathbf{v}$, the cosines measures of the query and the original data are given by

$$[0,\ 0.6667,\ 0.7746,\ 0.3333,\ 0.3333]^\top$$

which shows that Doc 2 and Doc 3 are most relevant.

Orthogonality in $\mathbb{C}^n$

Let $\mathbf{u} := [u_1, \ldots, u_n]^\top$ and $\mathbf{v} := [v_1, \ldots, v_n]^\top$ be vectors in $\mathbb{C}^n$. Then recall that

$$\langle \mathbf{u}, \mathbf{v} \rangle = u_1 \bar{v}_1 + \cdots + u_n \bar{v}_n = \mathbf{v}^* \mathbf{u} \text{ and } |\langle \mathbf{u}, \mathbf{v} \rangle| = \|\mathbf{u}\| \, \|\mathbf{v}\| \cos\theta, \text{ where } \theta \in [0, \pi/2].$$

If $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ then $\mathbf{u}$ and $\mathbf{v}$ are called mutually orthogonal and is written as $\mathbf{u} \perp \mathbf{v}$.

Definition: A set of vectors $S := \{\mathbf{u}_1, \ldots, \mathbf{u}_m\} \subset \mathbb{C}^n$ is called an orthogonal set if the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are mutually orthogonal, that is, $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$ for all $i \neq j$. If S is an orthogonal set then the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are called orthogonal vectors.

If S is an orthogonal set and $\|\mathbf{u}_j\| = 1$ for $j = 1 : m$ then S is called an orthonormal set (ONS) and the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$ are called orthonormal vectors. ■

Example: The standard unit vectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$ in $\mathbb{R}^n$ are orthonormal vectors. The vectors $\mathbf{u}_1 := [2, 1, -1]^\top, \mathbf{u}_2 := [0, 1, 1]^\top, \mathbf{u}_3 := [1, -1, 1]^\top$ in $\mathbb{R}^3$ are orthogonal vectors.

Orthonormal basis

Fact: If $S := \{\mathbf{u}_1, \ldots, \mathbf{u}_m\} \subset \mathbb{C}^n$ is an orthonormal set then S is linearly independent.

Proof: $c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m = \mathbf{0} \Longrightarrow c_j = \mathbf{u}_j^*(c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m) = 0$ for $j = 1 : m$. ■

Definition: Let $\mathcal{V}$ be a subspace of $\mathbb{C}^n$ and $\mathcal{B} := \{\mathbf{u}_1, \ldots, \mathbf{u}_m\} \subset \mathcal{V}$. Then $\mathcal{B}$ is called an orthonormal basis (ONB) of $\mathcal{V}$ if $\mathcal{B}$ is an orthonormal set and $\operatorname{span}(\mathcal{B}) = \mathcal{V}$.

If $\mathcal{B}$ is an orthogonal set and is a basis of $\mathcal{V}$ then $\mathcal{B}$ is called an orthogonal basis of $\mathcal{V}$. ■

Example: The vectors $\mathbf{u}_1 := [2, 1, -1]^\top, \mathbf{u}_2 := [0, 1, 1]^\top, \mathbf{u}_3 := [1, -1, 1]^\top$ in $\mathbb{R}^3$ are orthogonal and linearly independent. Hence $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ is an orthogonal basis of $\mathbb{R}^3$.

Theorem: Let $\mathcal{V}$ be a subspace of $\mathbb{C}^n$ and $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ be an ONB of $\mathcal{V}$. Let $\mathbf{v} \in \mathcal{V}$. Then $\mathbf{v}$ can be expressed uniquely as

$$\mathbf{v} = \langle \mathbf{v}, \mathbf{u}_1 \rangle \mathbf{u}_1 + \cdots + \langle \mathbf{v}, \mathbf{u}_m \rangle \mathbf{u}_m = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{bmatrix}^* \mathbf{v}.$$

Proof: There exist unique scalars $c_1, \ldots, c_m$ in $\mathbb{C}$ such that $\mathbf{v} = c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m$ $\Longrightarrow \mathbf{u}_j^*\mathbf{v} = \mathbf{u}_j^*(c_1\mathbf{u}_1 + \cdots + c_m\mathbf{u}_m) = c_j \Longrightarrow c_j = \mathbf{u}_j^*\mathbf{v} = \langle \mathbf{v}, \mathbf{u}_j \rangle$ for $j = 1 : m$. ■

Unitary and orthogonal matrices

Definition: A matrix $U \in \mathbb{C}^{n\times n}$ is called unitary if $U^*U = UU^* = I_n$. A matrix $V \in \mathbb{C}^{m\times n}$ is called an isometry if $V^*V = I_n$. A matrix $Q \in \mathbb{R}^{n\times n}$ is called an orthogonal matrix if $Q^\top Q = QQ^\top = I_n$.

Remark: A matrix $Q \in \mathbb{R}^{n\times n}$ is orthogonal if and only if $Q^\top = Q^{-1}$.

Fact: A matrix $U \in \mathbb{C}^{m\times n}$ is an isometry $\iff$ columns of U are orthonormal.

Proof: If $U := \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_n \end{bmatrix}$ then $U^*U = [\mathbf{u}_i^*\mathbf{u}_j]_{n\times n} = I_n \iff \langle \mathbf{u}_j, \mathbf{u}_i\rangle = \mathbf{u}_i^*\mathbf{u}_j = \delta_{ij}$. ■

Example: The rotation matrix $A := \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is an orthogonal matrix.

Theorem: Let $U \in \mathbb{C}^{n\times n}$. Then the following statements are equivalent.

(a) U is unitary.

(b) $\|U\mathbf{x}\| = \|\mathbf{x}\|$ for all $\mathbf{x} \in \mathbb{C}^n$.

(c) $\langle U\mathbf{x}, U\mathbf{y}\rangle = \langle \mathbf{x}, \mathbf{y}\rangle$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{C}^n$.

Orthogonal subspaces in $\mathbb{R}^n$

Definition: Two subspaces $\mathcal{X}$ and $\mathcal{Y}$ of $\mathbb{R}^n$ are said to be orthogonal if $\mathbf{y}^\top \mathbf{x} = 0$ for all $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$. We write $\mathcal{X} \perp \mathcal{Y}$ when $\mathcal{X}$ and $\mathcal{Y}$ are orthogonal. In particular, we write $\mathbf{x} \perp \mathcal{Y}$ when $\mathbf{y}^\top \mathbf{x} = 0$ for all $\mathbf{y} \in \mathcal{Y}$. ■

Consider $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ in $\mathbb{R}^3$. Let $\mathcal{X} := \operatorname{span}(\mathbf{e}_1, \mathbf{e}_2)$ and $\mathcal{Y} := \operatorname{span}(\mathbf{e}_3)$. Then $\mathcal{X} \perp \mathcal{Y}$.

Let $A \in \mathbb{R}^{m \times n}$ and let $A^\top = \begin{bmatrix} \mathbf{y}_1 & \cdots & \mathbf{y}_m \end{bmatrix}$. Then $A\mathbf{x} = \begin{bmatrix} \mathbf{y}_1^\top \\ \vdots \\ \mathbf{y}_m^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{y}_1^\top \mathbf{x} \\ \vdots \\ \mathbf{y}_m^\top \mathbf{x} \end{bmatrix}$.

Consider $N(A) := \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\} \subset \mathbb{R}^n$ and $R(A) := \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\} \subset \mathbb{R}^m$. Then $\mathbf{x} \in N(A) \iff A\mathbf{x} = \mathbf{0} \iff \mathbf{y}_j^\top \mathbf{x} = 0$ for $j = 1 : m \iff \mathbf{x} \perp R(A^\top)$.

Fact: Let $A \in \mathbb{R}^{m \times n}$. Then $N(A)$ and $R(A^\top)$ are mutually orthogonal subspaces of $\mathbb{R}^n$, that is, $N(A) \perp R(A^\top)$. Similarly, $N(A^\top) \perp R(A)$.

Orthogonal Decomposition Theorem

Definition: Let $\mathcal{X}$ be a subspace of $\mathbb{R}^n$. Define $\mathcal{X}^\perp := \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} \perp \mathcal{X}\}$. The set $\mathcal{X}^\perp$ is called the orthogonal complement of $\mathcal{X}$. ■

Fact: If $\mathcal{X}$ is a subspace of $\mathbb{R}^n$ then $\mathcal{X}^\perp$ is a subspace of $\mathbb{R}^n$ and $\mathcal{X} \cap \mathcal{X}^\perp = \{\mathbf{0}\}$.

Theorem: Let $\mathcal{X}$ be a subspace of $\mathbb{R}^n$ and let $\mathbf{v} \in \mathbb{R}^n$. Then there exist unique $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{X}^\perp$ such that $\mathbf{v} = \mathbf{x} + \mathbf{y}$. Equivalently, $\mathbb{R}^n = \mathcal{X} \oplus \mathcal{X}^\perp$.

Proof: Let $\{\mathbf{u}_1, \ldots, \mathbf{u}_m\}$ be an orthonormal basis of $\mathcal{X}$. Let $\mathbf{v} \in \mathbb{R}^n$. Define $\mathbf{x} := \langle \mathbf{v}, \mathbf{u}_1 \rangle \mathbf{u}_1 + \cdots + \langle \mathbf{v}, \mathbf{u}_m \rangle \mathbf{u}_m$ and $\mathbf{y} := \mathbf{v} - \mathbf{x}$. Then $\mathbf{v} = \mathbf{x} + \mathbf{y}$ and $\mathbf{x} \in \mathcal{X}$.

Note that $\langle \mathbf{y}, \mathbf{u}_j \rangle = \langle \mathbf{v}, \mathbf{u}_j \rangle - \langle \mathbf{x}, \mathbf{u}_j \rangle = 0$ for $j = 1 : m \Longrightarrow \mathbf{y} \perp \mathcal{X} \Longrightarrow \mathbf{y} \in \mathcal{X}^\perp$.
Thus $\mathbf{v} = \mathbf{x} + \mathbf{y}$ with $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{X}^\perp$. Since $\mathcal{X} \cap \mathcal{X}^\perp = \{\mathbf{0}\}$, the result follows. ■

Fact: Let $\mathcal{X}$ be a supspace of $\mathbb{R}^n$. Then $(\mathcal{X}^\perp)^\perp = \mathcal{X}$.

Proof: $\mathcal{X} \perp \mathcal{X}^\perp \Longrightarrow \mathcal{X} \subset (\mathcal{X}^\perp)^\perp$. Let $\mathbf{v} \in (\mathcal{X}^\perp)^\perp$. By projection theorem, $\mathbf{v} = \mathbf{x} + \mathbf{y}$ with $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{X}^\perp \Longrightarrow \mathbf{y} \perp \{\mathbf{x}, \mathbf{v}\} \Longrightarrow 0 = \mathbf{y}^\top \mathbf{v} = \mathbf{y}^\top \mathbf{x} + \mathbf{y}^\top \mathbf{y} = \mathbf{y}^\top \mathbf{y} \Longrightarrow \mathbf{y} = \mathbf{0}$.
Hence $\mathbf{v} = \mathbf{x} \in \mathcal{X} \Longrightarrow (\mathcal{X}^\perp)^\perp \subset \mathcal{X}$. ■

Fundamental Subspace Theorem

Remark: The orthogonal decomposition theorem is also called Projection Theorem.

Theorem: Let $A \in \mathbb{R}^{m\times n}$. Then $N(A)^\perp = R(A^\top)$ and $N(A^\top) = R(A)^\perp$. Further,

$$\begin{aligned} \mathbb{R}^n &= N(A) \oplus R(A^\top) \text{ and } N(A) \perp R(A^\top), \\ \mathbb{R}^m &= N(A^\top) \oplus R(A) \text{ and } N(A^\top) \perp R(A). \end{aligned}$$

Proof: We have seen that $N(A) \perp R(A^\top)$ which imples that $N(A) \subset R(A^\top)^\perp$. Now, $\mathbf{x} \in R(A^\top)^\perp \Longrightarrow \mathbf{x} \perp R(A^\top) \Longrightarrow \mathbf{x} \perp A^\top \mathbf{e}_j$ for $j = 1 : m \Longrightarrow (A^\top \mathbf{e}_j)^\top \mathbf{x} = \mathbf{e}_j^\top A\mathbf{x} = \mathbf{0}$ for $j = 1 : m \Longrightarrow A\mathbf{x} = \mathbf{0} \Longrightarrow \mathbf{x} \in N(A) \Longrightarrow R(A^\top)^\perp \subset N(A)$.

This proves $N(A) = R(A^\top)^\perp$. Now replacing A with $A^\top$ yields $N(A^\top) = R(A)^\perp$. Finally, by orthogonal decomposition theorem

$$\begin{aligned} \mathbb{R}^n &= N(A) \oplus N(A)^\perp = N(A) \oplus (R(A^\top)^\perp)^\perp = N(A) \oplus R(A^\top), \\ \mathbb{R}^m &= N(A^\top) \oplus N(A^\top)^\perp = N(A^\top) \oplus (R(A)^\perp)^\perp = N(A^\top) \oplus R(A). \blacksquare \end{aligned}$$

Remark: The subspaces $R(A), N(A), R(A^\top)$ and $N(A^\top)$ are called four fundamental subspaces of an $m \times n$ matrix A.

Orthogonalization

Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be linearly independent vectors in $\mathbb{C}^n$ such that $\mathbf{v}_2^*\mathbf{v}_1 \neq 0$. We wish to construct orthonormal vectcors $\mathbf{u}_1$ and $\mathbf{u}_2$ such that

$$\text{span}(\mathbf{v}_1) = \text{span}(\mathbf{u}_1) \text{ and } \text{span}(\mathbf{v}_1, \mathbf{v}_2) = \text{span}(\mathbf{u}_1, \mathbf{u}_2).$$

Set $\mathbf{u}_1 := \dfrac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$. Then $\text{span}(\mathbf{v}_1) = \text{span}(\mathbf{u}_1)$. Next, choose $\alpha \in \mathbb{C}$ such that $(\mathbf{v}_2 - \alpha\mathbf{u}_1) \perp \mathbf{u}_1$. This gives $\langle \mathbf{v}_2 - \alpha\mathbf{u}_1, \mathbf{u}_1\rangle = 0 \Longrightarrow \langle \mathbf{v}_2, \mathbf{u}_1\rangle = \alpha$. Thus $(\mathbf{v}_2 - \langle \mathbf{v}_2, \mathbf{u}_1\rangle\mathbf{u}_1) \perp \mathbf{u}_1$.

Define $\mathbf{u}_2 := \dfrac{\mathbf{v}_2 - \langle \mathbf{v}_2, \mathbf{u}_1\rangle\mathbf{u}_1}{\|\mathbf{v}_2 - \langle \mathbf{v}_2, \mathbf{u}_1\rangle\mathbf{u}_1\|}$. Then $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthonormal and $\mathbf{u}_1, \mathbf{u}_2 \in \text{span}(\mathbf{v}_1, \mathbf{v}_2)$.
Now

$$\mathbf{v}_2 = (\|\mathbf{v}_2 - \langle \mathbf{v}_2, \mathbf{u}_1\rangle\mathbf{u}_1\|)\mathbf{u}_2 + \langle \mathbf{v}_2, \mathbf{u}_1\rangle\mathbf{u}_1 \in \text{span}(\mathbf{u}_1, \mathbf{u}_2) \Longrightarrow \mathbf{v}_1, \mathbf{v}_2 \in \text{span}(\mathbf{u}_1, \mathbf{u}_2).$$

This shows that $\text{span}(\mathbf{v}_1, \mathbf{v}_2) = \text{span}(\mathbf{u}_1, \mathbf{u}_2)$.

If $\mathbf{v}_3$ is another vector then defne $\mathbf{u}_3 := \dfrac{\mathbf{v}_3 - \langle \mathbf{v}_3, \mathbf{u}_1\rangle\mathbf{u}_1 - \langle \mathbf{v}_3, \mathbf{u}_2\rangle\mathbf{u}_2}{\|\mathbf{v}_3 - \langle \mathbf{v}_3, \mathbf{u}_1\rangle\mathbf{u}_1 - \langle \mathbf{v}_3, \mathbf{u}_2\rangle\mathbf{u}_2\|}$. Then $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ are orthonormal and $\text{span}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) = \text{span}(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$.

Gram-Schmidt Orthogonalization

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be linearly indepedent vectors in $\mathbb{C}^n$. Then there exist orthonormal vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ in $\mathbb{C}^n$ such that

$$\operatorname{span}(\mathbf{v}_1, \ldots, \mathbf{v}_j) = \operatorname{span}(\mathbf{u}_1, \ldots, \mathbf{u}_j) \text{ for } j = 1 : m.$$

The Gram-Schmidt process constructs orthonormal vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ as follows. Define

$$\begin{aligned} \mathbf{u}_1 &:= \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}, \\ \mathbf{u}_j &:= \frac{\mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{u}_1 \rangle \mathbf{u}_1 - \cdots - \langle \mathbf{v}_j, \mathbf{u}_{j-1} \rangle \mathbf{u}_{j-1}}{\|\mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{u}_1 \rangle \mathbf{u}_1 - \cdots - \langle \mathbf{v}_j, \mathbf{u}_{j-1} \rangle \mathbf{u}_{j-1}\|}, \ j = 2 : m. \end{aligned}$$

Note that $\|\mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{u}_1 \rangle \mathbf{u}_1 - \cdots - \langle \mathbf{v}_j, \mathbf{u}_{j-1} \rangle \mathbf{u}_{j-1}\| \neq 0 \iff$ the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_j$ are linearly independent. By induction $\operatorname{span}(\mathbf{v}_1, \ldots, \mathbf{v}_j) = \operatorname{span}(\mathbf{u}_1, \ldots, \mathbf{u}_j)$ for $j = 1 : n$.

QR factorization

Setting $r_{11} := \|\mathbf{v}_1\|$, $r_{jj} := \|\mathbf{v}_j - \langle \mathbf{v}_j, \mathbf{u}_1\rangle \mathbf{u}_1 - \cdots - \langle \mathbf{v}_j, \mathbf{u}_{j-1}\rangle \mathbf{u}_{j-1}\|$ and $r_{kj} := \langle \mathbf{v}_j, \mathbf{u}_k\rangle$, for $k = 1 : j-1$, we have

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{u}_1 r_{11}, \\ \mathbf{v}_j &= \langle \mathbf{v}_j, \mathbf{u}_1\rangle \mathbf{u}_1 + \cdots + \langle \mathbf{v}_j, \mathbf{u}_{j-1}\rangle \mathbf{u}_{j-1} + r_{jj}\mathbf{u}_j, \\ &= \mathbf{u}_1 r_{1j} + \cdots + \mathbf{u}_{j-1} r_{j-1,j} + r_{jj}\mathbf{u}_j, \quad j = 2 : m. \end{aligned}$$

Then, in matrix notation, we have

$$A := \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_m \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ & r_{22} & \cdots & r_{2m} \\ & & \ddots & \vdots \\ & & & r_{mm} \end{bmatrix} = QR.$$

Thus, if $A \in \mathbb{C}^{n\times m}$ and $\operatorname{rank}(A) = m$, then A has a QR factorization $A = QR$, where Q is an isometry and R is upper triangular and nonsingular.

Example

Consider $\mathbf{v}_1 := \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}^\top$, $\mathbf{v}_2 := \begin{bmatrix} 2 & 1 & 0 \end{bmatrix}^\top$ and $\mathbf{v}_3 := \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^\top$. Then by the Gram-Schmidt process, we have $r_{11} := \|\mathbf{v}_1\| = \sqrt{2}$ which gives

$$\mathbf{u}_1 := \frac{\mathbf{v}_1}{r_{11}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Next, we have $r_{12} := \mathbf{u}_1^\top \mathbf{v}_2 = \sqrt{2}$ and

$$\begin{aligned} \mathbf{q}_2 &:= \mathbf{v}_2 - (\mathbf{u}_1^\top \mathbf{v}_2)\mathbf{u}_1 = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} - \frac{\sqrt{2}}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \\ \mathbf{u}_2 &:= \frac{\mathbf{q}_2}{r_{22}} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \text{ where } r_{22} := \|\mathbf{q}_2\| = \sqrt{3}. \end{aligned}$$

Example

Finally, $r_{13} := \mathbf{u}_1^\top \mathbf{v}_3 = 1/\sqrt{2}$ and $r_{23} := \mathbf{u}_2^\top \mathbf{v}_3 = 0$. Hence we have

$$\mathbf{q}_3 := \mathbf{v}_3 - (\mathbf{u}_1^\top \mathbf{v}_3)\mathbf{u}_1 - (\mathbf{u}_2^\top \mathbf{v}_3)\mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \frac{1}{2}\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix},$$

$$\mathbf{u}_3 := \frac{\mathbf{q}_3}{r_{33}} = \frac{1}{\sqrt{6}}\begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}, \quad \text{where } r_{33} := \|\mathbf{q}_3\| = \frac{\sqrt{6}}{2}.$$

Setting $A := \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix}$ and $Q := \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \end{bmatrix}$, we have the QR factorization of A

$$\underbrace{\begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{6}} \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} \sqrt{2} & \sqrt{2} & \frac{1}{\sqrt{2}} \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \frac{\sqrt{6}}{2} \end{bmatrix}}_{R}. \blacksquare$$

MA580H Matrix Computations

Lectures 5 & 6: System of Linear Equations-I

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Solution of triangular system
- Gaussian elimination
- LU decomposition

Linear system

Let $A \in \mathbb{R}^{n\times n}$ be nonsingular and $b \in \mathbb{R}^n$.

Problem: Solve $Ax = b$ for $x \in \mathbb{R}^n$.

Idea: For a nonsingular M, the solution of $MAx = Mb$ is given by

$$x = (MA)^{-1}Mb = A^{-1}M^{-1}Mb = A^{-1}b.$$

So, the strategy is to choose M so that the system

$$MAx = Mb$$

is easy to solve. Gaussian elimination provides such an M for which MA is upper triangular.

The MATLAB command

```
>> x= A\b
```

solves the system $Ax = b$ using Gaussian elimination.

Lower triangular linear system

Consider the lower triangular linear system of equations

$$\begin{bmatrix} \ell_{11} & & & \\ \ell_{21} & \ell_{22} & & \\ \vdots & \vdots & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

By forward substitution, we have

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_i &= \left(b_i - \textstyle\sum_{j=1}^{i-1} \ell_{ij} x_j\right)/\ell_{jj}, \quad i = 2:n. \end{aligned}$$

Cost: n^2 flops

Indeed, $\sum_{i=1}^{n} 2i = \int_0^n 2x dx +$ lower order terms $\simeq n^2$.

Column-oriented forward substitution

Writing $Lx = b$ as

$$\mathtt{L(:,1)x(1)} + \cdots + \mathtt{L(:,n)x(n)} = \mathtt{b}$$

we obtain column-oriented forward substitution.

```
x = zeros(n,1);
for j=1:n-1
    x(j) =b(j)/L(j,j);
    b(j+1:n) = b(j+1:n)-L(j+1:n,j)*x(j);
end
x(n) = b(n)/L(n,n);
```

- Solving a lower triangular system costs n^2 flops.

Upper triangular linear system

Consider the upper triangular system

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

If $u_{11}, \ldots, u_{nn}$ are nonzero, then by back substitution, we have a unique solution

$$\begin{aligned} x_n &= b_n / u_{nn} \\ x_i &= \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) / u_{jj}, \quad i = n-1, \ldots, 1. \end{aligned}$$

Cost: An upper triangular system is solved by back substitution and costs n^2 flops.

Gaussian elimination

Strategy: Transform a given linear system $Ax = b$ to an equivalent triangular linear system $\widehat{A}x = \widehat{b}$. Consider the system

$$\begin{aligned} x - y - z &= 2 \\ 3x - 3y + 2z &= 16 \\ 2x - y + z &= 9 \end{aligned} \iff \underbrace{\left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 3 & -3 & 2 & 16 \\ 2 & -1 & 1 & 9 \end{array}\right]}_{\text{augmented matrix}}$$

Use first equation to eliminating x from 2nd and 3rd equation

$$\begin{aligned} x - y - z &= 2 \\ 5z &= 10 \\ y + 3z &= 5 \end{aligned} \iff \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 0 & 5 & 10 \\ 0 & 1 & 3 & 5 \end{array}\right].$$

Now interchange 2nd and 3rd equations

$$\begin{aligned} x - y - z &= 2 \\ y + 3z &= 5 \\ 5z &= 10 \end{aligned} \iff \left[\begin{array}{ccc|c} 1 & -1 & -1 & 2 \\ 0 & 1 & 3 & 5 \\ 0 & 0 & 5 & 10 \end{array}\right] \Rightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}.$$

Gaussian elimination

```
function x=gauss(A,b)
% x=gauss(A,b) solves the linear system Ax=b using Gaussian
% elimination with partial pivoting on [A, b].
n=length(b);
norma=norm(A,1);
A=[A,b]; % augmented matrix
for i=1:n
[maxpv,kmax]=max(abs(A(i:n,i))); % look for Pivot A(kmax,i)
kmax=kmax+i-1;
if maxpv < 1e-14*norma; % only small pivots
error('matrix is singular')
end
if i ~= kmax % interchange rows
A([i, kmax],:) = A([kmax, i], :);
end
A(i+1:n,i)=A(i+1:n,i)/A(i,i); % elimination step
A(i+1:n,i+1:n+1)=A(i+1:n,i+1:n+1)-A(i+1:n,i)*A(i,i+1:n+1);
end
x=backsubs(A,A(:,n+1));
```

Gaussian elimination (GE)

Gaussian elimination can be rewritten as a method that factorizes a matrix. We consider three variants of GE. These variants yield three matrix factorizations, namely,

- LU factorization: $\mathrm{A} = \mathrm{LU}$
- Row permuted LU factorization: $\mathrm{PA} = \mathrm{LU}$
- Row and column permuted LU factorization: $\mathrm{PAQ} = \mathrm{LU}$

Here P and Q are permutation matrices. An $n \times n$ permutation matrix is obtained by permuting rows of the identity matrix I_n.

The matrix L is unit lower triangular and U is upper triangular. A lower triangular matrix L is called unit lower triangular if the diagonal entries of L are 1, that is, $\ell_{jj} = 1$ for $j = 1 : n$.

LU Decomposition

Definition: An LU decomposition of a matrix $A \in \mathbb{R}^{n\times n}$ is a factorization of the form $A = LU$, where L is unit lower triangular and U is upper triangular. Thus

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & \\ \vdots & \ddots & \\ \ell_{n1} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ & \ddots & \vdots \\ & & u_{nn} \end{bmatrix} = LU.$$

We wish to construct a nonsingular M such that MA is upper triangular. We expect M to have the following properties:

- $M = L_{n-1}^{-1} L_{n-2}^{-1} \cdots L_1^{-1}$
- Each L_j is unit lower triangular
- The product $L := L_1 L_2 \cdots L_{n-1}$ requires NO computation

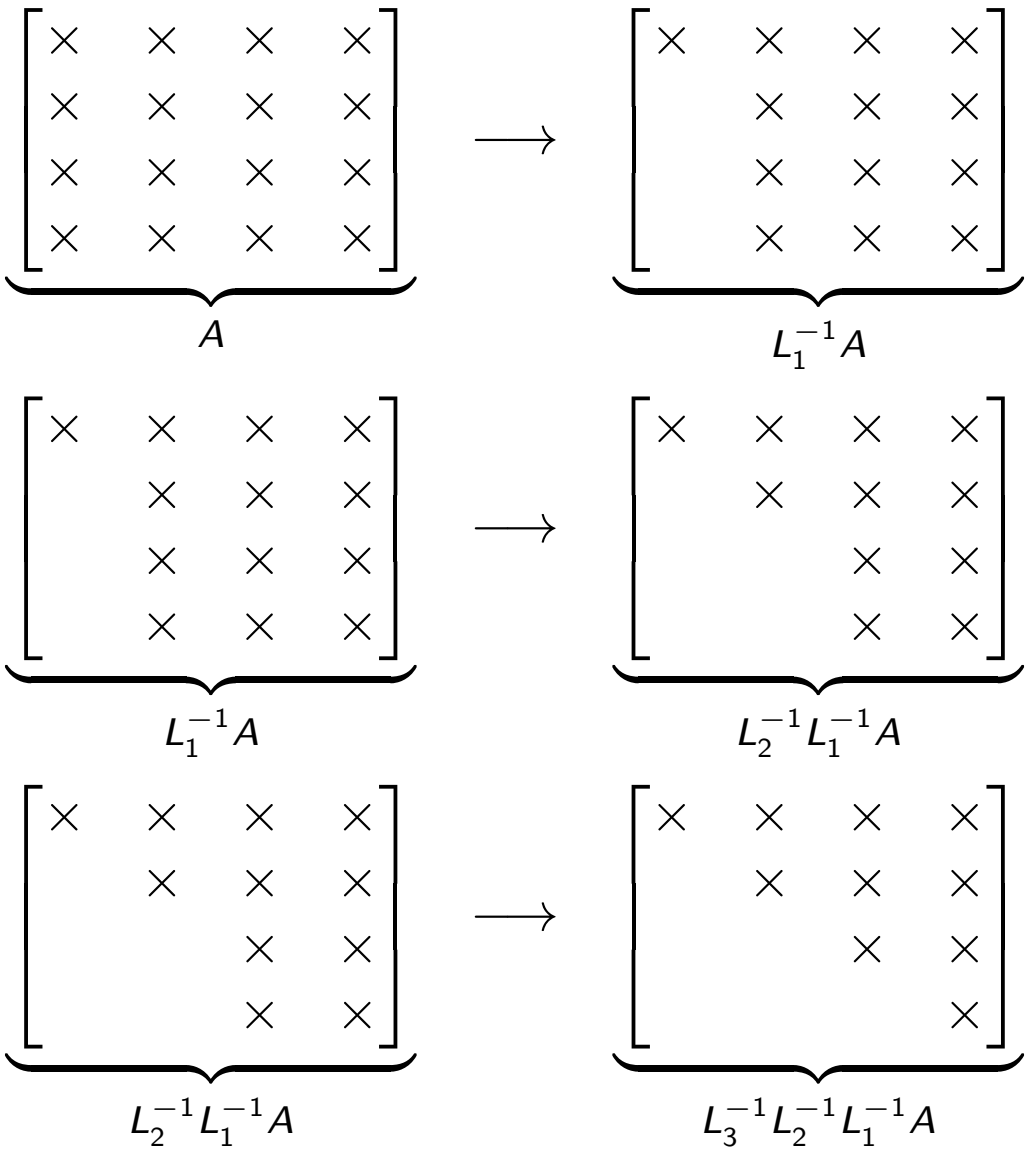
Then

$$MA = U \Longrightarrow L_{n-1}^{-1} L_{n-2}^{-1} \cdots L_1^{-1} A = U \Longrightarrow A = LU,$$

where L is unit lower-triangular and U is upper-triangular.

LU factorization

Suppose A is 4×4 matrix. Then schematically



Example

Let $A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix}$. Consider $L_1 := \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 0 & 1 \end{bmatrix}$. Then

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -7 & 0 & 1 \end{bmatrix} \text{ and } L_1^{-1}A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -21 \end{bmatrix}.$$

Now consider $L_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$. Then $L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$,

$$U := L_2^{-1}L_1^{-1}A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & -9 \end{bmatrix} \text{ and } L := L_1L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix}.$$

Thus we obtain $A = LU$.

Elimination matrix

Define $\ell_k := \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{nk} \end{bmatrix}$ and $L_k := I + \ell_k e_k^\top$ for $k = 1 : (n-1)$.

Then

$$L_k = I + \begin{bmatrix} 0 & \cdots & 0 & \ell_k & 0 & \cdots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \ell_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & \ell_{nk} & & & 1 \end{bmatrix}$$

is unit lower triangular.

Product of elimination matrices

Consider $L_k = I + \ell_k e_k^\top$. By construction $e_k^\top \ell_k = 0$. Consequently

$$\underbrace{(I + \ell_k e_k^\top)}_{L_k}(I - \ell_k e_k^\top) = I + \ell_k e_k^\top - \ell_k e_k^\top - \ell_k e_k^\top \ell_k e_k^\top = I.$$

This shows that $L_k^{-1} = I - \ell_k e_k^\top$. Next observe that

$$L_k L_{k+1} = (I + \ell_k e_k^\top)(I + \ell_{k+1} e_{k+1}^\top) = I + \ell_k e_k^\top + \ell_{k+1} e_{k+1}^\top.$$

Consequently

$$\begin{aligned} L &= L_1 L_2 \cdots L_{n-1} = I + \ell_1 e_1^\top + \ell_2 e_2^\top + \cdots + \ell_{n-1} e_{n-1}^\top \\ &= I + \begin{bmatrix} \ell_1 & \ell_2 & \cdots & \ell_{n-1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{n,n-1} & 1 \end{bmatrix}. \end{aligned}$$

Creating zeros via elimination matrix

Applying L_k^{-1} to the k-th column of an $n \times n$ matrix A, we have

$$L_k^{-1} \begin{bmatrix} a_{1k} \\ \vdots \\ a_{kk} \\ a_{k+1,k} \\ \vdots \\ a_{nk} \end{bmatrix} = (I - \ell_k e_k^\top) \begin{bmatrix} a_{1k} \\ \vdots \\ a_{kk} \\ a_{k+1,k} \\ \vdots \\ a_{nk} \end{bmatrix} = \begin{bmatrix} a_{1k} \\ \vdots \\ a_{kk} \\ a_{k+1,k} \\ \vdots \\ a_{nk} \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{nk} \end{bmatrix} a_{kk}$$

$$= \begin{bmatrix} a_{k1} \\ \vdots \\ a_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{when } \ell_{ik} = a_{ik}/a_{kk},\ i = k+1 : n.$$

This shows that if $a_{kk} \neq 0$ then L_k can be used to create zeros in the k-th column of A below a_{kk}.

Multiplying a matrix by an elimination matrix

Let $A \in \mathbb{R}^{n\times n}$ and $L_k := I + \ell_k e_k^\top \in \mathbb{R}^{n\times n}$. Then

$$L_k^{-1}A = (I - \ell_k e_k^\top)A = A - \ell_k e_k^\top A = A - \ell_k \begin{bmatrix} a_{k1} & \cdots & a_{kn} \end{bmatrix}$$

$$= \left[\begin{array}{ccccc} a_{11} & \cdots & a_{kk} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ a_{k1} & \cdots & a_{kk} & \cdots & a_{kn} \\ \hline a_{k+1,n} & \cdots & a_{k+1,k} & \cdots & a_{k+1,n} \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nk} & \cdots & a_{nn} \end{array}\right] - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \ell_{k+1,k} \\ \vdots \\ \ell_{nk} \end{bmatrix} \begin{bmatrix} a_{k1} & \cdots & a_{kn} \end{bmatrix}.$$

The outer product shows that the first k rows of A remain unchanged when L_k^{-1} is multiplied to the left of A. Let $B := L_k^{-1}A$. In MATLAB, B can be written compactly as a rank-1 update (outer product from)

$$\texttt{B} = \texttt{A}(\texttt{k}+1:\texttt{n},:) - \ell(\texttt{k}+1:\texttt{n}) * \texttt{A}(\texttt{k},:)$$

Gaussian elimination = LU decomposition

For $L_2 := I + \ell_2 e_2^\top$ with $\ell_{i2} := a_{i2}^{(1)}/a_{22}^{(1)}$, $i = 3:n$, we have

$$L_2^{-1}L_1^{-1}A = \left[\begin{array}{c|c|ccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ \hline 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ \hline 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{array}\right],$$

where $a_{ij}^{(2)} = a_{ij}^{(1)} - \ell_{i2}a_{2j}^{(1)}$.
Cost: $2(n-2)^2$ flops.

The elimination is possible only when $a_{22}^{(1)} \neq 0$. The vector ℓ_2 can be stored in the second column of $L_2^{-1}L_1^{-1}A$ in place of zeros.

Again, overwriting A, in MATLAB notation, we have

$$\texttt{A(3:n,2)} = \texttt{A(3:n,2)/A(2,2);} \quad \texttt{\% multipliers}$$
$$\texttt{A(3:n,3:n)} = \texttt{A(3:n,3:n)} - \texttt{A(3:n,2)} * \texttt{A(2,3:n);}$$

Hence we have $L_{n-1}^{-1}\cdots L_1^{-1}A = U \Rightarrow A = L_1L_2\cdots L_{n-1}U = LU$.
Cost: $2(n-1)^2 + 2(n-2)^2 + \cdots + 2 \simeq 2n^3/3$ flops.

Example

Consider $A := \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$. Then

$$L_1 = I + \begin{bmatrix} 0 \\ 2 \\ -1 \end{bmatrix} e_1^\top, \quad L_1^{-1}A = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix},$$

$$L_2 = I + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e_2^\top, \quad L_2^{-1}L_1^{-1}A = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix}.$$

This gives

$$A = L_1 L_2 \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 0 & 4 \end{bmatrix}.$$

Gaussian Elimination with No Pivoting (GENP)

```
function [L, U] = GENP(A);
% [L U] = GENP(A) produces a unit
% lower triangular matrix L and an upper
% triangular matrix U so that A= LU.

[n, n] = size(A);
for k = 1:n-1
    % compute multipliers for k-th step
    A(k+1:n,k) = A(k+1:n,k)/A(k,k);
    % update A(k+1:n,k+1:n)
    j = k+1:n;
    A(j,j) = A(j,j)-A(j,k)*A(k,j);
end
% strict lower triangle of A, plus I
L = eye(n,n)+ tril(A,-1);
U = triu(A); % upper triangle of A
```

Solution of $Ax = b$ by LU factorization

An $n \times n$ linear system $Ax = b$ can be solved in three steps:

- Compute LU factorization $A = LU$. Cost: $\frac{2n^3}{3}$ flops.
- Solve $Ly = b$ for y. Cost: n^2 flops.
- Solve $Ux = y$ for x. Cost: n^2 flops.

Thus the cost for solving system $Ax = b$ is $2n^3/3$ flops.

Question: What will be the complexity if the system $Ax = b$ is solved as $\mathtt{x} = \mathtt{A}^{-1} * \mathtt{b}$? Answer: $2n^3/3 + 2n^3$ flops.

Question: Does LU decomposition of A exist when A is nonsingular?

Answer: Not always. LU decomposition of $A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$ does not exist.

Theorem: Let A be nonsingular. Then A admits a unique LU factorization $\Leftrightarrow$ all leading principal submatrices of A are nonsingular, that is, $A(1:j, 1:j)$ is nonsingular for $j = 1:n$.

Existence of LU factorization

Proof: Suppose that $A = LU$ exists and unique. Then writing

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

we have $\det(A_{11}) = \det(L_{11})\det(U_{11}) = \det(U_{11}) \neq 0$. (Why?)

Conversely, suppose that all leading principal submatrices of A are nonsingular. We prove the result by induction. Suppose the result is true for $n-1$.

Let $\hat{A} = A(1:n-1, 1:n-1)$ and $\hat{A} = \hat{L}\hat{U}$ be unique LU factorization. Then writing

$$A = \begin{bmatrix} \hat{A} & b \\ c & a_{nn} \end{bmatrix} = \begin{bmatrix} \hat{L} & 0 \\ \ell & 1 \end{bmatrix} \begin{bmatrix} \hat{U} & u \\ 0 & d \end{bmatrix} = \begin{bmatrix} \hat{L}\hat{U} & \hat{L}u \\ \ell\hat{U} & \ell u + d \end{bmatrix},$$

we have $\hat{L}u = b$, $\ell\hat{U} = c$ and $d = a_{nn} - \ell u$ which give unique ℓ, u, d.

Finally, $0 \neq \det(A) = \det(\hat{U})d \Rightarrow d \neq 0$. This completes the proof. ■

MA580H Matrix Computations

Lecture 7: System of Linear Equations-II

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Gaussian elimination with pivoting
- Permutated LU decomposition

Pivoting

Consider $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. Since the pivot element, that is, $(1,1)$ entry of the matrix is 0, the elimination fails to reduce the matrix to upper triangular form.

However, interchanging the rows we obtain an upper triangular matrix

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{P}\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}}_{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{L}\underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}}_{U}.$$

The matrix P is a permutation matrix. A permutation matrix is obtained by interchanging rows of identity matrix. The process of interchanging rows is called partial pivoting

Theorem (GEPP): Let A be an $n \times n$ matrix. Then there is a permutation matrix P such that

$$PA = LU$$

where L is unit lower triangular and U is upper triangular.

Gaussian elimination with partial pivoting

Let P_1 be a permutation matrix so that $(1,1)$ entry of P_1A is nonzero. Then for $L_1 := I + \ell_1 e_1^\top$, with $\ell_{i1} := a_{i1}/a_{11}$, $i = 2:n$, we have

$$L_1^{-1}P_1A = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{array}\right], \text{ where } a_{ij}^{(1)} = a_{ij} - \ell_{i1}a_{1j}. \textbf{ Cost: } 2(n-1)^2 \text{ flops.}$$

If $a_{22}^{(1)} = 0$ then elimination breaks down.However, if say $a_{n2}^{(1)} \neq 0$ then we can interchange rows and make $a_{n2}^{(1)}$ as the pivot element and continue elimination.

$$P_2L_1^{-1}P_1A = \left[\begin{array}{c|ccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \end{array}\right].$$

Here P_2 is the permutation matrix that interchanges second row with n-th row of $L_1^{-1}A$.

Gaussian elimination with partial pivoting

For $L_2 := I + \ell_2 e_2^\top$ with $\ell_{i2} := a_{i2}^{(1)}/a_{n2}^{(1)}$, $i = 3:n$, we have

$$L_2^{-1}P_2L_1^{-1}P_1A = \left[\begin{array}{c|c|ccc} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ \hline 0 & a_{n2}^{(1)} & a_{n3}^{(1)} & \cdots & a_{nn}^{(1)} \\ \hline 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{array}\right], \quad a_{ij}^{(2)} = a_{ij}^{(1)} - \ell_{i2}a_{2j}^{(1)}. \textbf{ Cost: } 2(n-2)^2 \text{ flops.}$$

Repeating the process we have $L_{n-1}^{-1}P_{n-1}L_{n-2}^{-1}\cdots P_2L_1^{-1}P_1A = U$.
Cost: $2(n-1)^2 + 2(n-2)^2 \cdots + 2 \simeq 2n^3/3$ flops.

The matrix $L_{n-1}^{-1}P_{n-1}L_{n-2}^{-1}\cdots P_2L_1^{-1}P_1$ may NOT be lower triangular. However, we show that

$$PA = \underbrace{\hat{L}_1\hat{L}_2\cdots\hat{L}_{n-2}L_{n-1}}_{L}U = LU,$$

where L is unit lower triangular, $\hat{L}_j$'s are obtained from L_j's by permutating their multipliers and $P := P_{n-1}P_{n-2}\cdots P_2P_1$.

GEPP (cont.)

```
    if (A(k,k) ~= 0)
        % compute multipliers for k-th step
        A(k+1:n,k) = A(k+1:n,k)/A(k,k);
        % update A(k+1:n,k+1:n)
        j = k+1:n;
        A(j,j) = A(j,j)-A(j,k)*A(k,j);
    end
end
% strict lower triangle of A, plus I
L = eye(n,n)+ tril(A,-1);
U = triu(A); % upper triangle of A
```

The search for the largest entry in each column guarantees that the denominator `A(k,k)` in the entries `L(k+1:n,k) = A(k+1:n,k)/A(k,k)` is at least as large as the numerators.

This ensures that $|L(i,j)| \leq 1$ for all i, j. This is crucial for stability.

Example

Consider

$$\underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{P_1} \underbrace{\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}}_{A} = \begin{bmatrix} 4 & 9 & -3 \\ 2 & 4 & -2 \\ -2 & -3 & 7 \end{bmatrix}.$$

Then $L_1 = I + \begin{bmatrix} 0 \\ \frac{1}{2} \\ -\frac{1}{2} \end{bmatrix} e_1^\top$, $L_1^{-1}A = \begin{bmatrix} 4 & 9 & -3 \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{3}{2} & \frac{11}{2} \end{bmatrix}$. Now

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_2} \begin{bmatrix} 4 & 9 & -3 \\ 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{3}{2} & \frac{11}{2} \end{bmatrix} = \begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Then $L_2 = I + \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{3} \end{bmatrix} e_2^\top$, $L_2^{-1}P_2L_1^{-1}P_1A = \begin{bmatrix} 4 & 9 & -3 \\ 0 & \frac{3}{2} & \frac{11}{2} \\ 0 & 0 & \frac{4}{3} \end{bmatrix}$

Example (cont.)

Thus we have $L_2^{-1}P_2L_1^{-1}P_1A = U \Longrightarrow A = MU$, where $M := (L_2^{-1}P_2L_1^{-1}P_1)^{-1} = P_1L_1P_2L_2$. Now

$$P_1L_1 = P_1\begin{bmatrix}1 & 0 & 0\\ \frac{1}{2} & 1 & 0\\ -\frac{1}{2} & 0 & 1\end{bmatrix} = \begin{bmatrix}\frac{1}{2} & 1 & 0\\ 1 & 0 & 0\\ -\frac{1}{2} & 0 & 1\end{bmatrix}$$

$$P_2L_2 = P_2\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -\frac{1}{3} & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ 0 & -\frac{1}{3} & 1\\ 0 & 1 & 0\end{bmatrix}$$

shows that M is not lower triangular. Next, observe that

$$U = L_2^{-1}P_2L_1^{-1}P_1A = L_2^{-1}P_2L_1^{-1}P_2P_2P_1A \Longrightarrow P_2P_1A = P_2L_1P_2L_2U = LU$$

where $L := P_2L_1P_2L_2$ is unit lower triangular. Indeed

$$P_2L_1P_2L_2 = \begin{bmatrix}1 & 0 & 0\\ -\frac{1}{2} & 1 & 0\\ \frac{1}{2} & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -\frac{1}{3} & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\\ -\frac{1}{2} & 1 & 0\\ \frac{1}{2} & -\frac{1}{3} & 1\end{bmatrix}.$$

Permuted LU decomposition (PA = LU)

By GEPP we have $L_{n-1}^{-1}P_{n-1}\cdots L_2^{-1}P_2L_1^{-1}P_1A = U$, where P_k is the permutation matrix and $L_k := I + \ell_k e_k^\top$ is the elimination matrix at the k-th step.

Theorem: Set $L(\ell_k) := L_k$. Then

$$P_{n-1}\cdots P_{k+1}L(\ell_k) = L(P_{n-1}\cdots P_{k+1}\ell_k)P_{n-1}\cdots P_{k+1}.$$

Set $\widehat{L}_k := L(P_{n-1}\cdots P_{k+1}\ell_k)$. Then $\widehat{L}_k$ is unit lower triangular and

$$L_{n-1}^{-1}P_{n-1}\cdots L_2^{-1}P_2L_1^{-1}P_1A = L_{n-1}^{-1}\widehat{L}_{n-2}^{-1}\cdots\widehat{L}_2^{-1}\widehat{L}_1^{-1}P_{n-1}\cdots P_1A.$$

Thus, setting $P := P_{n-1}P_{n-2}\cdots P_1$ and $L := \widehat{L}_1\cdots\widehat{L}_{n-2}L_{n-1}$, we have $PA = LU$.

Proof: The first $m-1$ rows of P_m (P_m is used at the m-th step of elimination) are the same as the first $m-1$ rows of I_n. Hence $e_k^\top P_m = e_k^\top$ for $k = 1 : m-1$.

Permuted LU decomposition (PA = LU)

Since $e_k^\top P_m = e_k^\top$ for $k = 1 : m-1$, we have
$P_m L(\ell_k) = P_m(I + \ell_k e_k^\top) = P_m + P_m \ell_k e_k^\top = P_m + P_m \ell_k e_k^\top P_m = L(P_m \ell_k) P_m.$

Consequently, $P_{n-1} \cdots P_{k+1} L(\ell_k) = L(P_{n-1} \cdots P_{k+1} \ell_k) P_{n-1} \cdots P_{k+1}$.
Now

$$\begin{aligned} L_3^{-1} P_3 L_2^{-1} P_2 L_1^{-1} P_1 A &= L(-\ell_3) P_3 L(-\ell_2) P_2 L(-\ell_1) P_1 A \\ &= L(-\ell_3) L(-P_3 \ell_2) P_3 P_2 L(-\ell_1) P_1 A \\ &= L(-\ell_3) L(-P_3 \ell_2) L(-P_3 P_2 \ell_1) P_3 P_2 P_1 A. \end{aligned}$$

Continuing this process, we have

$$L_{n-1}^{-1} P_{n-1} \cdots L_2^{-1} P_2 L_1^{-1} P_1 A = L_{n-1}^{-1} \widehat{L}_{n-2}^{-1} \cdots \widehat{L}_2^{-1} \widehat{L}_1^{-1} P_{n-1} \cdots P_1 A.$$

Hence the results follow. ■

Solution of linear system using GEPP

A linear system $Ax = b$ can be solved using GEPP as follows.

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

1. Compute $PA = LU$ ($2n^3/3$ flops)
2. Compute $y = Pb$ (permute the entries of b, no arithmetic needed)
3. Solve $Lz = y$ by forward substitution (n^2 flops)
4. Solve $Ux = z$ by back substitution (n^2 flops).

Total Cost: $\frac{2n^3}{3}$ flops.

GEPP is the standard method used in practice for solving a linear system. GEPP is a default method in MATLAB for solution of a linear system. The command `x = A\b` solves $Ax = b$ using GEPP. The command `[L, U, P] = lu(A)` computes $PA = LU$.

Gaussian elimination with complete pivoting

Gaussian elimination with complete pivoting (GECP) is a variant of GE. At the k-step, GECP searches not just column `A(k:n,k)` but the entire submatrix `A(k:n,k:n)` for the largest entry and then swaps rows and columns to put that entry into `A(k,k)`. After $(k-1)$ steps

$$L_{k-1}^{-1}P_{k-1}\cdots L_1^{-1}P_1AQ_1\cdots Q_{k-1} = \left[\begin{array}{ccc|ccc} * & \cdots & * & * & \cdots & * \\ & \ddots & \vdots & \vdots & \cdots & \vdots \\ & & * & * & \cdots & * \\ \hline & & & a_{kk} & \cdots & a_{kn} \\ & & & \vdots & \ddots & \vdots \\ & & & a_{nk} & \cdots & a_{nn} \end{array}\right].$$

After $n-1$ steps, we have $PAQ = LU$ where P and Q are permutations matrices. If $\mathrm{rank}(A) = r$ then $U = \begin{bmatrix} U_1 & U_2 \\ 0 & 0 \end{bmatrix}$, where U_1 is an $r \times r$ nonsingular upper triangular matrix.

GECP guarantees $|L(i,j)| <= 1$ and $|U(i,j)| <= |U(i,i)|$.

Cost: $2n^3/3 + n^3/3 = n^3$ flops. Additional $n^3/3$ flops is due to finding maximum element at each step.

GEPP versus GECP

- GECP is more expensive ($\mathcal{O}(n^3)$ more operations) than GEPP.
- GECP is usually no more accurate than GEPP which is why GEPP is a default method.
- Examples exist for which GECP does much better than GEPP.
- GEPP and GECP work extremely well in the presence of roundoff.

***

MA580H Matrix Computations

Lecture 8: Cholesky Factorization

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Characterization of positive definite matrices
- Cholesky factorization

Quadratic forms

A pure quadratic $f(x, y)$ comes directly from a symmetric 2 by 2 matrix!

$$\mathbf{x}^\top A\mathbf{x} \text{ in } \mathbb{R}^2 \quad ax^2 + 2bxy + cy^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & b \\ b & c \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

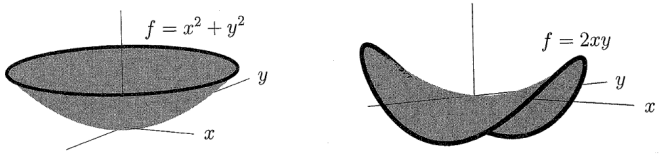


Figure 6.1: A bowl and a saddle: Definite $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and indefinite $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

For any symmetric matrix A, the product $\mathbf{x}^\top A\mathbf{x}$ is a pure quadratic form $f(x_1, \ldots, x_n)$:

$$\mathbf{x}^\top A\mathbf{x} \text{ in } \mathbb{R}^n \quad \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_n \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j.$$

Positive definite matrices

A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be

- positive semidefinite if $x^\top Ax \geq 0$ for all $x \in \mathbb{R}^n$ (written as $A \succeq 0$)
- positive definite if $x^\top Ax > 0$ for all nonzero $x \in \mathbb{R}^n$ (written as $A \succ 0$)

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be

- positive semidefinite if $x^*Ax \geq 0$ for all $x \in \mathbb{C}^n$ (written as $A \succeq 0$)
- positive definite if $x^*Ax > 0$ for all nonzero $x \in \mathbb{C}^n$ (written as $A \succ 0$)

A real positive definite matrix is also referred to as a symmetric positive definite (SPD) matrix.

Remark: Let $A \in \mathbb{C}^{n\times n}$. Then $x^*Ax \in \mathbb{R}$ for all $x \in \mathbb{C}^n \iff A = A^*$.
But $A \in \mathbb{R}^{n\times n}$ and $x^\top Ax \in \mathbb{R}$ for all $x \in \mathbb{R}^n \not\Longrightarrow A = A^\top$.

Indeed, if $A = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$ then $x^\top Ax = (x_1 + x_2)^2 \geq 0$ for all $x \in \mathbb{R}^2$ but $A \neq A^\top$.

Positive definite matrices

If $A \in \mathbb{R}^{n\times n}$ is partitioned in the form

$$A = \left[\begin{array}{c|c} A_m & B \\ \hline C & D \end{array}\right], \quad A_m \in \mathbb{R}^{m\times m},$$

then A_m is called a principal submatrix of A. Note that

$$A^\top = A \iff A_m^\top = A_m, \quad C = B^\top, \quad D^\top = D.$$

It follows that if A is SPD then so is A_m. Indeed, for any nonzero $x \in \mathbb{R}^m$, we have

$$x^\top A_m x = \left[\begin{array}{c} x \\ \hline 0 \end{array}\right]^\top \left[\begin{array}{c|c} A_m & B \\ \hline C & D \end{array}\right] \left[\begin{array}{c} x \\ \hline 0 \end{array}\right] > 0.$$

In particular, if A is SPD then $a_{jj} = e_j^\top A e_j > 0$ for $j = 1 : n$. Also, A is nonsingular (why?).

Positive definite matrices

Facts: Let $A \in \mathbb{R}^{n\times n}$ be an SPD matrix. Then the following results hold:

1. If $X \in \mathbb{R}^{n\times p}$ with $\operatorname{rank}(X) = p$ then $X^\top AX$ is SPD. Indeed, for all nonzero $y \in \mathbb{R}^p$,

$$Xy \neq 0 \quad \text{(why?) and} \quad y^\top (X^\top AX)y = (Xy)^\top A(Xy) > 0 \Longrightarrow X^\top AX \text{ is SPD.}$$

2. Leading principal submatrices of A are SPD, that is, $\mathtt{A}(1:\mathtt{j}, 1:\mathtt{j})$ is SPD for $j = 1:n$.
3. Let $A = \left[\begin{array}{c|c} A_m & B^\top \\ \hline B & D \end{array}\right]$. Then $S := D - BA_m^{-1}B^\top$ is the Schur complement of A_m. Now

$$\left[\begin{array}{c|c} A_m & B^\top \\ \hline B & D \end{array}\right] = \left[\begin{array}{c|c} I & 0 \\ \hline BA_m^{-1} & I \end{array}\right] \left[\begin{array}{c|c} A_m & 0 \\ \hline 0 & D - BA_m^{-1}B^\top \end{array}\right] \left[\begin{array}{c|c} I & 0 \\ \hline BA_m^{-1} & I \end{array}\right]^\top$$

shows that

$$A \text{ is SPD} \iff A_m \text{ and } S := D - BA_m^{-1}B^\top \text{ are SPD.}$$

LDV factorization

Theorem: Suppose that all leading principal submatrices $A \in \mathbb{R}^{n\times n}$ are nonsingular. Then $A = LDV$ is a unique decomposition of A, where L is unit lower triangular, D is diagonal, and V is unit upper triangular.

Proof: By assumption, A has a unique LU factorization $A = LU$. Let $D := \mathrm{diag}(u_{11}, \ldots, u_{nn})$, where $u_{11}, \ldots, u_{nn}$ are diagonal entries of U. Then $V := D^{-1}U$ is unit upper triangular and $A = LDV$. ■

Corollary: If $A \in \mathbb{R}^{n\times n}$ is symmetric and all leading principal submatrices of A are nonsingular then $A = LDL^\top$ is a unique factorization of A, where L is unit lower triangular and D is a diagonal matrix.

Corollary: If A is SPD then $A = LDL^\top$ is a unique factorization of A, where L is unit lower triangular and D is a diagonal SPD matrix.

Cholesky factorization

Theorem: Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. Then A is SPD $\iff$ $A = GG^\top$, where G is a unique lower triangular matrix with positive diagonal entries.

Proof: $A = GG^\top \Rightarrow x^\top Ax = x^\top GG^\top x = (G^\top x)^\top G^\top x = \|G^\top x\|_2 > 0$ for $x \neq 0 \Rightarrow A$ is SPD.

A is SPD $\Rightarrow A = LDL^\top$ is a unique factorization, where L is unit lower triangular and D is diagonal SPD matrix. Let D be given by $D = \mathrm{diag}(d_{11}, \ldots, d_{nn})$. Since $d_{jj} > 0$, define $\sqrt{D} := \mathrm{diag}(\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}})$ and $G := L\sqrt{D}$. Then $A = L\sqrt{D}(L\sqrt{D})^\top = GG^\top$. ■

Definition: If A is SPD then $A = GG^\top$, where G lower triangular with positive diagonals, is called the Cholesky factorization of A and G is called the Cholesky factor of A.

Example:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}^\top .$$

Algorithm (inner product)

Let $A := \begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{bmatrix}$ and $G := \begin{bmatrix} g_{11} & \\ g_{21} & g_{22} \end{bmatrix}$. Then $A = GG^\top$ yields

$$\begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} g_{11} & \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} g_{11} & g_{21} \\ & g_{22} \end{bmatrix} = \begin{bmatrix} g_{11}^2 & g_{11}g_{21} \\ g_{11}g_{21} & g_{21}^2 + g_{22}^2 \end{bmatrix}.$$

Equating the columns, we have

$$\begin{array}{l} a_{11} = g_{11}^2 \\ a_{21} = g_{11}g_{21} \\ a_{22} = g_{21}^2 + g_{22}^2 \end{array} \quad \Longrightarrow \quad \begin{array}{l} g_{11} = \sqrt{a_{11}} \\ g_{21} = a_{21}/g_{11} \\ g_{22} = \sqrt{a_{22} - g_{21}^2} \end{array}$$

Remark: The factorization is possible if $a_{11} > 0$ and $a_{22} - g_{21}^2 > 0$.

Algorithm (inner product)

More generally, equating columns on both sides of $A = GG^\top$, we have

$$\begin{bmatrix} a_{11} \\ \vdots \\ a_{n1} \end{bmatrix} = g_{11} \begin{bmatrix} g_{11} \\ \vdots \\ g_{n1} \end{bmatrix}, \quad \begin{bmatrix} a_{22} \\ \vdots \\ a_{n2} \end{bmatrix} = g_{21} \begin{bmatrix} g_{21} \\ \vdots \\ g_{n1} \end{bmatrix} + g_{22} \begin{bmatrix} g_{22} \\ \vdots \\ g_{n2} \end{bmatrix}$$

$$\begin{bmatrix} a_{jj} \\ \vdots \\ a_{nj} \end{bmatrix} = g_{j1} \begin{bmatrix} g_{j1} \\ \vdots \\ g_{n1} \end{bmatrix} + g_{j2} \begin{bmatrix} g_{j2} \\ \vdots \\ g_{n2} \end{bmatrix} + \cdots + g_{jj} \begin{bmatrix} g_{jj} \\ \vdots \\ g_{nj} \end{bmatrix}, \; j = 1 : n$$

Algorithm (Inner product):

For $j = 1 : n$

$$g_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2}$$

$$g_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} g_{ik} g_{jk} \right) / g_{jj}, \; i = j + 1 : n$$

end

Cost: $n^3/3$ flops - half the cost of GE.

Algorithm (inner product)

Example: Consider $\begin{bmatrix} 16 & -16 & 0 \\ -16 & 41 & -5 \\ 0 & -5 & 5 \end{bmatrix}$. Then

$$\begin{aligned} g_{11} &= \sqrt{a_{11}} = \sqrt{16} = 4, \quad g_{21} = \frac{a_{21}}{g_{11}} = \frac{-16}{4} = -4, \quad g_{31} = \frac{a_{31}}{g_{11}} = \frac{0}{4} = 0 \\ g_{22} &= \sqrt{a_{22} - g_{21}^2} = \sqrt{41 - 16} = 5, \quad g_{32} = \frac{a_{32} - g_{31}g_{21}}{g_{22}} = \frac{-5 - 0 \times (-4)}{5} = -1 \\ g_{33} &= \sqrt{a_{33} - g_{31}^2 - g_{32}^2} = \sqrt{5 - 0 - 1} = 2. \end{aligned}$$

Hence

$$\begin{bmatrix} 16 & -16 & 0 \\ -16 & 41 & -5 \\ 0 & -5 & 5 \end{bmatrix} = \begin{bmatrix} 4 & & \\ -4 & 5 & \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 4 & & \\ -4 & 5 & \\ 0 & -1 & 2 \end{bmatrix}^\top.$$

Algorithm (outer product)

Let $A \in \mathbb{R}^{n\times n}$ be SPD. Then $A = GG^\top$ can be written as

$$\left[\begin{array}{c|c} a_{11} & h^\top \\ \hline h & \widehat{A} \end{array}\right] = \left[\begin{array}{c|c} g_{11} & 0 \\ \hline g & \widehat{G} \end{array}\right] \left[\begin{array}{c|c} g_{11} & g^\top \\ \hline 0 & \widehat{G}^\top \end{array}\right].$$

Equating the blocks, we have

$$\begin{aligned} a_{11} = g_{11}^2 &\implies g_{11} = \sqrt{a_{11}} \\ h = g_{11} g &\implies g = h/g_{11} \\ \widehat{A} = gg^\top + \widehat{G}\widehat{G}^\top &\implies \widehat{A} - gg^\top = \widehat{G}\widehat{G}^\top \end{aligned}$$

```
For k = 1:n
  A(k,k) = sqrt(A(k,k));
  g = A(k+1:n,k)/A(k,k); A(k+1:n,k) = g;
  A(k+1:n, k+1:n) = A(k+1:n, k+1:n)- g*g';
end
```

Cost: $n^3/3$ flops - half the cost of GE.

Example

$$\left[\begin{array}{c|cc} 25 & 15 & -5 \\ \hline 15 & 18 & 0 \\ -5 & 0 & 11 \end{array}\right] = \left[\begin{array}{c|cc} g_{11} & 0 & 0 \\ \hline g_{21} & g_{22} & 0 \\ g_{31} & g_{32} & g_{33} \end{array}\right] \left[\begin{array}{c|cc} g_{11} & g_{21} & g_{31} \\ \hline 0 & g_{22} & g_{32} \\ 0 & 0 & g_{33} \end{array}\right]$$

$$= \left[\begin{array}{c|cc} 5 & 0 & 0 \\ \hline 3 & g_{22} & 0 \\ -1 & g_{32} & g_{33} \end{array}\right] \left[\begin{array}{c|cc} 5 & 3 & -1 \\ \hline 0 & g_{22} & g_{32} \\ 0 & 0 & g_{33} \end{array}\right]$$

Equating (2, 2) blocks, we have

$$\begin{bmatrix} 18 & 0 \\ 0 & 11 \end{bmatrix} - \begin{bmatrix} 3 \\ -1 \end{bmatrix} \begin{bmatrix} 3 & -1 \end{bmatrix} = \begin{bmatrix} 9 & 3 \\ 3 & 10 \end{bmatrix} = \begin{bmatrix} g_{22} & 0 \\ g_{32} & g_{33} \end{bmatrix} \begin{bmatrix} g_{22} & g_{32} \\ 0 & g_{33} \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 0 \\ 1 & g_{33} \end{bmatrix} \begin{bmatrix} 3 & 1 \\ 0 & g_{33} \end{bmatrix}$$

Equating (2, 2) entry, we have $10 - 1 = g_{33}^2 \Longrightarrow g_{33} = 3$.

Solving SPD system

Let $A \in \mathbb{R}^{n\times n}$ be SPD and $b \in \mathbb{R}^n$. Then the system $Ax = b$ can be solved using Cholesky factorization as follows.

- Compute Cholesky factorization $A = GG^\top$. Cost: $n^3/3$ flops.
- Solve the lower triangular system $Gy = b$. Cost: n^2 flops.
- Solve the upper triangular system $G^\top x = y$. Cost: n^2 flops.

The MATLAB command `chol` computes Cholesky factorization of a positive definite matrix A. More specifically, the commands

$$\mathtt{R} = \mathtt{chol(A)} \text{ and } \mathtt{L} = \mathtt{chol(A,'lower')}$$

compute an upper triangular matrix R and a lower triangular matrix L such that

$$\mathtt{A} = \mathtt{R}^\top\mathtt{R} \text{ and } \mathtt{A} = \mathtt{L}\mathtt{L}^\top$$

A direct proof of Cholesky factorization

Problem: Let $A = \left[\begin{array}{c|c} a_{11} & h^\top \\ \hline h & D \end{array}\right]$, where $h \in \mathbb{R}^{n-1}$, be SPD. Then the Schur complement $S := D - hh^\top/a_{11}$ is SPD. Now use

$$\begin{aligned}
\left[\begin{array}{c|c} a_{11} & h^\top \\ \hline h & D \end{array}\right] &= \left[\begin{array}{c|c} 1 & 0 \\ \hline h/a_{11} & I_{n-1} \end{array}\right] \left[\begin{array}{c|c} a_{11} & h^\top \\ \hline 0 & D - hh^\top/a_{11} \end{array}\right] \\
&= \left[\begin{array}{c|c} 1 & 0 \\ \hline h/a_{11} & I_{n-1} \end{array}\right] \left[\begin{array}{c|c} a_{11} & 0 \\ \hline 0 & D - hh^\top/a_{11} \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline h/a_{11} & I_{n-1} \end{array}\right]^\top \\
&= \left[\begin{array}{c|c} \sqrt{a_{11}} & 0 \\ \hline h/\sqrt{a_{11}} & I_{n-1} \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & D - hh^\top/a_{11} \end{array}\right] \left[\begin{array}{c|c} \sqrt{a_{11}} & 0 \\ \hline h/\sqrt{a_{11}} & I_{n-1} \end{array}\right]^\top
\end{aligned}$$

and induction on n to prove that Cholesky factorization $A = GG^\top$ exists and is unique.

***

MA580H Matrix Computations

Lecture 9: Perturbation analysis of linear systems

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Vector and matrix norms
- Perturbation analysis of linear systems
- Stability analysis of GEPP

Vector norms

Let $\mathcal{V}$ be a vector space over $\mathbb{C}$. Then a function $\|\cdot\| : \mathcal{V} \to \mathbb{R}$ is called a norm on $\mathcal{V}$ if it satisfies the three fundamental properties:

(a) Positive definiteness: $\|v\| \geq 0$ and $\|v\| = 0 \iff v = 0$.

(b) Positively homogeneous: $\|\alpha v\| = |\alpha| \, \|v\|$ for $\alpha \in \mathbb{C}$ and $v \in \mathcal{V}$.

(c) Triangle inequality: $\|u + v\| \leq \|u\| + \|v\|$ for $u, v \in \mathcal{V}$.

Example: Consider $\mathbb{C}^n$ and the vector norms given by

1-norm: $\|x\|_1 := |x_1| + \cdots + |x_n|$.

2-norm: $\|x\|_2 := \sqrt{|x_1|^2 + \cdots |x_n|^2}$.

∞-norm: $\|x\|_\infty := \max_{1 \leq j \leq n} |x_j|$.

Example:

$$\|[1, 1, 3, 5]^\top\|_1 = 10, \|[1, 1, 3, 5]^\top\|_2 = 6 \text{ and } \|[1, 1, 3, 5]^\top\|_\infty = 5.$$

Matrix norms

Let $A \in \mathbb{C}^{m\times n}$. Then $A : \mathbb{C}^n \longrightarrow \mathbb{C}^m$, $x \longmapsto Ax$, is a linear map. Suppose $\mathbb{C}^n$ and $\mathbb{C}^m$ are equipped with norms. Then

$$\|A\| := \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

defines a norm on $\mathbb{C}^{m\times n}$ and is called an induced matrix norm or a subordinate matrix norm.

For the identity matrix $\|Ix\| = \|x\|$ and hence $\|I\| = 1$. Note that

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

for all $x \in \mathbb{C}^n$.

A matrix norm is said to be sub-multiplicative if $\|AB\| \leq \|A\| \, \|B\|$ holds for all A and B. An induced matrix norm is submultiplicative. Indeed, we have

$$\|ABx\| \leq \|A\| \, \|Bx\| \leq \|A\| \, \|B\| \, \|x\| \Longrightarrow \|AB\| \leq \|A\| \, \|B\|.$$

Matrix norms

The norms $\|A\|_1$, $\|A\|_2$ and $\|A\|_\infty$ induced by 1-norm, 2-norm and ∞-norm are called 1-norm, 2-norm and ∞-norm of A, respectively. Also $\|A\|_2$ is called the spectral norm of A.

Theorem: Let A be an $m \times n$ matrix. Then

$$\begin{aligned}
\|A\|_1 &= \max_{\|x\|_1=1} \|Ax\|_1 = \max_{1\le j\le n} \|Ae_j\|_1 = \max_{1\le j\le n} \|A(:,j)\|_1 \\
\|A\|_2 &= \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\lambda_{\max}(A^*A)} \\
\|A\|_\infty &= \max_{\|x\|_\infty=1} \|Ax\|_\infty = \max_{1\le i\le m} \|e_i^\top A\|_1 = \max_{1\le i\le m} \|A(i,:)\|_1,
\end{aligned}$$

where $\lambda_{\max}(A^*A)$ denotes that largest eigenvalue of A^*A.

Proof: We have $Ax = x_1Ae_1 + \cdots + x_nAe_n \Rightarrow \|Ax\|_1 \le \max_{1\le j\le n} \|Ae_j\|_1 \|x\|_1$. This yields $\|A\|_1 \le \max_{1\le j\le n} \|Ae_j\|_1$. But $\|Ae_j\|_1 \le \|A\|_1$ for all $j = 1 : n$. Hence we have $\|A\|_1 = \max_{1\le j\le n} \|Ae_j\|_1$. ■

Example

Let $A := \begin{bmatrix} 2 & 2 & -4 \\ 1 & 1 & 5 \\ 1 & 3 & 6 \end{bmatrix}$. Then $\|A\|_1 = \max\left(\|Ae_1\|_1, \|Ae_2\|_1, \|Ae_3\|_1\right) = \max(4, 6, 15) = 15.$

We have $\|A\|_\infty = \max(\|e_1^\top A\|_1, \|e_2^\top A\|_1, \|e_3^\top A\|_1) = \max(8, 7, 10) = 10.$

The spectral norm of A is given by $\|A\|_2 = \sqrt{\lambda_{\max}(A^\top A)} = 8.9826.$

Condition number and non-singularity

If A is nonsingular then when is $A + \Delta A$ nonsingular?

Fact: If $\|\Delta A\| \, \|A^{-1}\| < 1$ or equivalently, $\dfrac{\|\Delta A\|}{\|A\|}\mathrm{cond}(A) < 1$, then $A + \Delta A$ is nonsingular, where $\mathrm{cond}(A) := \|A\| \, \|A^{-1}\|$.

Proof: If possible, suppose that $A + \Delta A$ is singular. Then there is a vector x such that $\|x\| = 1$ and $(A + \Delta A)x = 0$.

Then $x = -A^{-1}\Delta A x \Longrightarrow 1 = \|A^{-1}\Delta A x\| \leq \|A^{-1}\| \, \|\Delta A\|$, which is a contradiction. ■

Remark: There is a ΔA such that $\|\Delta A\| \, \|A^{-1}\| = 1$ and $A + \Delta A$ is singular. In other words, the relative distance to nearest singular matrix $\propto \dfrac{1}{\mathrm{cond}(A)}$.

Condition number

Definition: Let A be an $n \times n$ nonsingular matrix. Then $\mathrm{cond}(A) := \|A\| \, \|A^{-1}\|$ is called the condition number of A. If $\mathrm{cond}(A)$ is NOT too large then A is said to be well-conditioned. If $\mathrm{cond}(A)$ is large then A is said to be ill-conditioned.

Note that for a subordinate matrix norm, we have $\mathrm{cond}(A) = \|A\| \, \|A^{-1}\| \geq 1$.

Remark: The determinant $\det(A)$ is not a good measure of ill-conditioning of A.

$$A := 10^{-1} I_n \Longrightarrow \det(A) = 10^{-n} \text{ and } \mathrm{cond}(A) = 1.$$

$$B := \begin{bmatrix} 1 & 10^{10} \\ 0 & 1 \end{bmatrix} \Longrightarrow \det(B) = 1 \text{ and } \mathrm{cond}_\infty(B) = (1 + 10^{10})^2 \simeq 10^{20}.$$

Notice that columns A are orthogonal whereas columns of B are nearly linearly dependent. Indeed, $\cos\theta = \langle Be_1, \, Be_2 \rangle / \|Be_1\|_2 \|Be_2\|_2 = 10^{10}/\sqrt{1 + 10^{20}} \simeq 1$.

Sensitivity analysis of linear systems

Consider the linear system

$$\underbrace{\begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}}_{\text{Hilbert matrix H}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

The matrix H is known as a Hilbert matrix, and it is known to be notoriously ill-conditioned.

To see what this means, set $x := \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^\top \in \mathbb{R}^n$ and define $b := Hx$. Then x is the solution of $Hx = b$.

Now we use MATLAB to solve the linear system and compare the computed solution with the known solution x.

Sensitivity of solutions of Hilbert systems

```
>> xx = hilb(12)\b; Warning:  Matrix is close to singular or
badly scaled.  Results may be inaccurate.  RCOND = 2.602837e-17.
```

n	$\|x - xx\|_\infty$	cond(H)
4	.4130030e-12	2.837500e+04
6	.6964739e-09	2.907028e+07
8	.7311487e-07	3.387279e+10
10	.2047785e-03	3.535233e+13
12	.2476695e-00	3.841961e+16

This would appear to justify the predictions that as n increases, roundoff errors would accumulate and destroy all accuracy in the computed solution of a linear system!

The Hilbert matrix is SPD but the computed solutions differ drastically from true solutions. Is it the fault of the algorithm?

Perturbation of linear system-I

Theorem: Let A be nonsingular and $\mathrm{cond}(A) := \|A\|\,\|A^{-1}\|$. Consider the linear systems $Ax = b$ and $A\hat{x} = b + \Delta b$. Then

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \mathrm{cond}(A)\frac{\|\Delta b\|}{\|b\|}.$$

Moreover, the upper bound is attained for some Δb.

Proof: We have $\hat{x} - x = A^{-1}\Delta b \Longrightarrow \|x - \hat{x}\| \leq \|A^{-1}\|\,\|\Delta b\|$. Now $Ax = b \Longrightarrow \|b\| \leq \|A\|\,\|x\| \Longrightarrow 1/\|x\| \leq \|A\|/\|b\|$, which yields the bound. ■

Residual bound: Let $\hat{x} = \mathrm{ALG}(A, b)$. Then the residual $r := b - A\hat{x}$ yields $A\hat{x} = b - r = b + \Delta b$, where $\Delta b := -r$. Hence we have the residual bound $\dfrac{\|x - \hat{x}\|}{\|x\|} \leq \mathrm{cond}(A)\dfrac{\|r\|}{\|b\|}$.

Example

Consider $A := \begin{bmatrix} 1000 & 999 \\ 999 & 998 \end{bmatrix}$. Then $A^{-1} = \begin{bmatrix} -998 & 999 \\ 999 & -1000 \end{bmatrix}$.

Thus $\|A\|_\infty = \|A\|_1 = \|A^{-1}\|_\infty = \|A^{-1}\|_1 = 1999$. Hence $\mathrm{cond}_\infty(A) = \mathrm{cond}_1(A) = (1999)^2 = 3.996 \times 10^6$.

Observe that $A \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1999 \\ 1997 \end{bmatrix}$ and $A^{-1} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1997 \\ -1999 \end{bmatrix}$.

Set $b := \begin{bmatrix} 1999 \\ 1997 \end{bmatrix}$ and $\Delta b := 10^{-2} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$. Consider system $A\hat{x} = b + \Delta b$. Then $\hat{x} = x + A^{-1}\Delta b = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 19.97 \\ -19.99 \end{bmatrix}$. This shows that

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} = 19.99 = (1999)^2 \frac{10^{-2}}{1999} = \mathrm{cond}_\infty(A) \frac{\|\Delta b\|_\infty}{\|b\|_\infty}.$$

Perturbation of linear system-II

Theorem: Consider the systems $Ax = b$ and $(A + \Delta A)\hat{x} = b + \Delta b$. Suppose that A is nonsingular and $\|\Delta A\|\,\|A^{-1}\| < 1$. Then

$$\begin{aligned}\frac{\|x - \hat{x}\|}{\|x\|} &\leq \frac{\mathrm{cond}(A)}{1 - \frac{\|\Delta A\|}{\|A\|}\mathrm{cond}(A)}\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right) \\ &\lesssim \mathrm{cond}(A)\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right).\end{aligned}$$

Proof: We have
$\hat{x} - x = -A^{-1}(\Delta A\hat{x} - \Delta b) \Longrightarrow \|\hat{x} - x\| \leq \|A^{-1}\|(\|\Delta A\|\,\|\hat{x}\| + \|\Delta b\|)$. Now

$$\|\hat{x}\| \leq \|\hat{x} - x\| + \|x\| \Longrightarrow (1 - \|A^{-1}\|\,\Delta A\|)\|x - \hat{x}\| \leq \|A^{-1}\|(\|\Delta A\|\,\|x\| + \|\Delta b\|).$$

Now diving both sides by $\|x\|$ and using the fact that
$b = Ax \Longrightarrow \|b\| \leq \|A\|\,\|x\| \Longrightarrow \|b\|/\|x\| \leq \|A\|$, we obtain the bound. ■

Example

Consider $A := \begin{bmatrix} 1 & 1+\delta \\ 1-\delta & 1 \end{bmatrix}$, where $\delta > 0$. Then $A^{-1} = \frac{1}{\delta^2}\begin{bmatrix} 1 & -1-\delta \\ -1+\delta & 1 \end{bmatrix}$. Hence $\mathrm{cond}_\infty(A) = \frac{(2+\delta)^2}{\delta^2}$.

For $\delta := 10^{-2}$, we have $\mathrm{cond}_\infty(A) = (201)^2 = 40401$.

Consider the linear systems $\begin{bmatrix} 1 & 1.01 \\ 0.99 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2.01 \\ 1.99 \end{bmatrix}$ whose solution is $x = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top$ and $\begin{bmatrix} 1 & 1.01 \\ 1 & 1 \end{bmatrix}\hat{x} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$.

Then $\hat{x} = \begin{bmatrix} 2 & 0 \end{bmatrix}^\top$. Note that $\Delta A = 10^{-2} e_2 e_1^\top$ and $\Delta b = 10^{-2}\begin{bmatrix} 1 & 1 \end{bmatrix}^\top$. We have $\|x - \hat{x}\|_\infty / \|x\|_\infty = 1$.

MA580H Matrix Computations

Lecture 10: Stability Analysis of Gaussian Elimination

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Stability analysis of GEPP/GECP
- Accuracy of computed solutions

Backward stability

An algorithm is a function $\mathrm{ALG} : (X, \|\cdot\|) \longrightarrow (Y, \|\cdot\|)$ such that

- computation of $\mathrm{ALG}(\text{input})$ involves only a finite number of steps
- and each step performs a finite number of elementary arithmetic operations.

Let $S(d)$ be a solution of a problem with given data d and $\mathrm{ALG}(d)$ be the computed solution. Then the accuracy of the computed solution $\mathrm{ALG}(d)$ is measured by the (relative) error

$$\text{Error } = \frac{\|\mathrm{ALG}(d) - S(d)\|}{\|S(d)\|}$$

Definition: An algorithm ALG is said to be backward stable (stable) if

- $\mathrm{ALG}(d) = S(d + \Delta d)$ for some $\Delta d \in X$ such that $\dfrac{\|\Delta d\|}{\|d\|} = \mathcal{O}(\mathbf{u})$.

The quantity $\dfrac{\|\Delta d\|}{\|d\|}$ is called the backward error.

Examples

Example 1: Consider $Ax = b$. Then $x = S(A, b) = A^{-1}b$. Let $\hat{x} = \mathrm{ALG}(A, b)$.Then

ALG stable $\Longrightarrow \hat{x} = \mathrm{ALG}(A, b) = S(A + \Delta A, b + \Delta b)$, that is, $(A + \Delta A)\hat{x} = b + \Delta b$ such that $\dfrac{\|\Delta A\|}{\|A\|} = \mathcal{O}(\mathbf{u})$ and $\dfrac{\|\Delta b\|}{\|b\|} = \mathcal{O}(\mathbf{u})$.

Example 2: Consider the LU decomposition $A = LU$. Let $[L, U] = \mathrm{ALG}(A)$. Then ALG stable $\Longrightarrow A + \Delta A = LU$ for some ΔA such that $\|\Delta A\|/\|A\| = \mathcal{O}(\mathbf{u})$.

Example 3: Suppose $\mathrm{ALG}(d)$ computes $f(d) = e^d$ for $d \in \mathbb{R}$. Then ALG is stable if $\mathrm{ALG}(d) = f(d + \Delta d) = e^{d+\Delta d}$ and $|\Delta d|/|d| = \mathcal{O}(\mathbf{u})$.

Accuracy

Backward stability of ALG guarantees
$\mathrm{ALG}(d) = S(d + \Delta d)$ and $\|\Delta d\|/\|d\| = \mathcal{O}(\mathbf{u})$.

What can be said about the error in the solution?

$$\begin{aligned} \text{Error} &= \frac{\|\mathrm{ALG}(d) - S(d)\|}{\|S(d)\|} = \frac{\|S(d + \Delta d) - S(d)\|}{\|S(d)\|} \\ &\leq \kappa_S(d) \frac{\|\Delta d\|}{\|d\|}. \end{aligned}$$

- The quantity $\kappa_S(d)$ is called the condition number of S at d and measures the sensitivity of S at d.
- The algorithm ALG has no control on $\kappa_S(d)$.

Ill-conditioning

- For small relative changes in d we have

$$\frac{\|S(d+\Delta d)-S(d)\|}{\|S(d)\|} \lesssim \kappa_S(d)\frac{\|\Delta d\|}{\|d\|}$$

$$\begin{pmatrix}\text{Error in}\\ \text{solution}\end{pmatrix} \lesssim \text{cond.} \times \begin{pmatrix}\text{Error in}\\ \text{data}\end{pmatrix}$$

- Thus $S(d)$ is ill-conditioned if $\kappa_S(d) \gg 1$. Otherwise, the problem is well-conditioned.
- How large $\kappa_S(d)$ is large enough? The answer depends on how choosy you are!
- If $\kappa_S(d) = 10^s$ then s digits may be lost in the solution computed by a stable algorithm.

Estimating the condition number

If S is differentiable at d then

$$\kappa_S(d) \simeq \frac{\|J_S(d)\|\,\|d\|}{\|S(d)\|},$$

where $J_S(d) = \left[\frac{\partial S_i}{\partial x_j}(d)\right]$ is the Jacobian of S at d.

Example: Consider $S(d) = \sqrt{d}$. Then $J_S(d) = S'(d) = 1/(2\sqrt{d})$, for $d \neq 0$ and $\mathrm{cond}_S(d) = 1/2$. ■

Example: Consider $S(d_1, d_2) = d_1 - d_2$. Then $J_S(d) = [1, -1]$ and

$$\kappa_S(d) = \frac{2\|d\|_\infty}{|d_1 - d_2|}.$$

For $d_1 := 1$, and $d_2 := 1 - 10^{-5}$, $\kappa_S(d) = 2 \times 10^5$. ■

Wilkinson's result (1961)

Theorem: Suppose we solve $Ax = b$ using GEPP in floating point arithmetic with unit roundoff $\mathbf{u}$. Let $\hat{x}$ be the computed solution. Then

$$(A + \Delta A)\hat{x} = b \text{ and } \frac{\|\Delta A\|_\infty}{\|A\|_\infty} \leq 2n^3 g_{\text{pp}}(A)\mathbf{u}$$

where $g_{\text{pp}}(A)$ is the pivot growth given by

$$g_{\text{pp}}(A) := \frac{\max_{ij} |U(i,j)|}{\max_{ij} |A(i,j)|} = \frac{\|U\|_{\max}}{\|A\|_{\max}}$$

Thus, $\|x - \hat{x}\|_\infty / \|x\|_\infty \lesssim 2n^3 g_{\text{pp}}(A)\text{cond}_\infty(A)\mathbf{u}$.

- Elegant way of accounting for rounding errors. Bounds backward error rather than the error.
- Draws attention to pivot growth factor g_{pp}.
- Both $g_{\text{pp}}(A)$ and $\text{cond}_\infty(A)$ are easy to compute after getting L and U, costing just an extra $\mathcal{O}(n^2)$ flops.

Growth factor for GEPP

What do we know about $g_{\rm pp}(A)$?

Wilkinson (1954) proved that $g_{\rm pp}(A) \leq 2^{n-1}$. Usually $g_{\rm pp}(A) \simeq 1$ in practice. But examples exists for which $g_{\rm pp}(A) = 2^{n-1}$.

Wilkinson's matrix: 5×5 Wilkinson's matrix W is given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 \\ -1 & -1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2^2 \\ 0 & 0 & 0 & 1 & 2^3 \\ 0 & 0 & 0 & 0 & 2^4 \end{bmatrix}.$$

Note that $g_{\rm pp}(W) = 2^4$.

For an $n \times n$ Wilkinson matrix W, we have $W = LU$ with $U(n, n) = 2^{n-1}$. Hence $g_{\rm pp}(W) = 2^{n-1}$. The matrix W can be generated in MATLAB as follows

```
W = tril( 2*eye(n)-ones(n) ); W(:, n) = ones(n,1);
```

Growth factor for GEPP

An $n \times n$ matrix A is said to be diagonally dominant if $|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|$ for $i = 1 : n$.

An $n \times n$ matrix A is said to be banded with bandwidth ℓ if $a_{ij} = 0$ for all $|i - j| > \ell$. For example, if $\ell = 1$ then A is tridiagonal and if $\ell = 2$ then A is pentadiagonal.

An $n \times n$ matrix A is said to be Hessenberg (i.e., upper Hessenberg form) if $a_{ij} = 0$ for $i > j + 1$.

Special matrices:

Matrix	$g_{\mathrm{pp}}(A)$
diag. dom	2
tridiagonal	2
banded (bandwidth p)	$2^{2p-1} - (p-1)2^{p-2}$
Hessenberg	n
SPD	1

Growth factor for GECP

- Wilkinson (1961) proved

$$g_{\rm cp}(A) \leq n^{1/2}(2.3^{1/2}\cdots n^{1/2})^{1/2} \sim cn^{1/2}n^{\frac{1}{4}\log n}.$$

- Usually, in practice, $g_{\rm cp}(A) \sim 1$. Determining the largest possible value of $g_{\rm cp}(A)$ is still an open problem.

Remark: There is no correlation between pivot growth of A and the condition number of A, that is, no correlation between $\mathrm{PG}(A)$ and $\mathrm{cond}(A)$. This is illustrated by Golub matrix.

```
function A = golub(n)
s = 10;
L = tril(round(s*randn(n)),-1)+eye(n);
U = triu(round(s*randn(n)),1)+eye(n);
A = L*U;
```

Golub matrix

`A = golub(10)` gives

$$\begin{bmatrix} 1 & -21 & 29 & -4 & 0 & 5 & -3 & -13 & -14 & -2 \\ 18 & -377 & 530 & -80 & -3 & 90 & -62 & -257 & -247 & -38 \\ -23 & 490 & -610 & 20 & -39 & -115 & 2 & 124 & 349 & 29 \\ 9 & -190 & 269 & -283 & -288 & 37 & -170 & -315 & -262 & -64 \\ 3 & -56 & 148 & -177 & -23 & 257 & -828 & -353 & 46 & -34 \\ -13 & 271 & -383 & -78 & -216 & -176 & 298 & 122 & 60 & 8 \\ -4 & 83 & -117 & -85 & -134 & -72 & -39 & -63 & -117 & -62 \\ 3 & -48 & 204 & -92 & 39 & 143 & -189 & -314 & 247 & -89 \\ 36 & -742 & 1159 & -290 & -127 & 176 & 267 & -747 & -358 & -291 \\ 28 & -574 & 916 & -113 & 164 & 397 & -289 & -552 & -333 & 414 \end{bmatrix}$$

For $n = 10$, we have $g_{pp}(A) = 1$ and $\mathrm{cond}_\infty(A) = 2.9219 \times 10^{18}$. For Wilkinson matrix with $n = 50$, we have $g_{pp}(A) = 2^{49} = 5.6295 \times 10^{14}$ and $\mathrm{cond}(A) = 22.306$.

Remark: Pivot growth for Cholesky factorization is 1. Hence the algorithm is backward stable.

MA580H Matrix Computations

Least-Squares Problem(LSP)

Lecture 11: Method of Normal Equation

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- Least squares problem
- Method of Normal Equation
- Tikhonov regularization

Overdetermined linear systems

- Overdetermined system: We have considered a linear system $Ax = b$ only when A is a square matrix. We now consider an $m \times n$ linear system $Ax = b$ when $m > n$. This is called an overdetermined linear system.
- Underdetermined system: The complementary underdetermined case $m < n$ turns up less frequently and will NOT be discussed.
- Least-squares solution: An overdetermined system is mostly inconsistent and hence does not have a solution. So, we need to define what is a best possible solution. There are multiple useful options but we consider only the least squares solution in which 2-norm of the residual vector is minimized.

Linear and nonlinear least squares problems

Problem statement: Given data points $(x_1, y_1), \ldots, (x_m, y_m)$, determine a function $f(x, \alpha)$ that best fit the data, where $\alpha := [\alpha_1, \ldots, \alpha_n]^\top$ is a parameter vector that solves the minimization problem

$$\min_{\alpha} \sum_{j=1}^{m} (f(x_j, \alpha) - y_j)^2.$$

Linear least-squares problem: In this case $f(x, \alpha)$ is linear with respect to α, that is, $f(x, \alpha) := \alpha_1 \phi_1(x) + \cdots + \alpha_n \phi_n(x)$, where $\phi_1(x), \ldots, \phi_n(x)$ are given model functions.

For example, $f(x, \alpha) := \alpha_1 + \alpha_2 x$ leads to a linear regression problem (straightline fit).

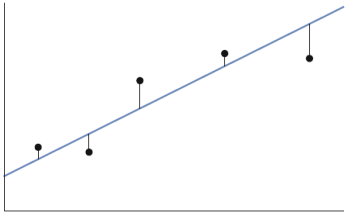
Nonlinear least-squares problem: In this case $f(x, \alpha)$ is nonlinear with respect to α.

For example, $f(x, \alpha) := \dfrac{1}{1 + e^{-(\alpha_1 x + \alpha_2)}}$ leads to a logistic regression problem.

Certain nonlinear least squares problem can be transformed to linear least squares problem by change of variables. For example, $f(x, \alpha) := \alpha_1 e^{\alpha_2 x} \Longrightarrow \log f(x, \alpha) = \log \alpha_1 + \alpha_2 x = c_1 + c_2 x$.

Linear regression

Given data points $(t_1, b_1), \ldots, (t_m, b_m)$ in $\mathbb{R}^2$, find a straight line $f(t, \alpha) := \alpha_1 + \alpha_2 t$ that best fit the data. The task is to minimize the error $\sum_{j=1}^{m} (f(t_j, \alpha) - b_j)^2$ for all $\alpha_1, \alpha_2 \in \mathbb{R}$.



Setting $r_i := f(t_i, \alpha) - b_i \implies f(t_i, \alpha) = b_i + r_i \implies \alpha_1 + \alpha_2 t_i = b_i + r_i$ for $i = 1 : m$. This yields the LSP

$$\begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} \implies \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

Linear regression

Now consider the LSP

$$Ax = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = b.$$

Set $\mu_t := (t_1 + \cdots + t_m)/m, \sigma_t^2 := (t_1^2 + \cdots + t_m^2)/m, \mu_b := (b_1 + \cdots + b_m)/m$ and $\sigma_{tb} := (t_1 b_1 + \cdots + t_m b_m)/m$. Then the normal equation $A^\top Ax = A^\top b$ gives

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_1 & t_2 & \cdots & t_m \end{bmatrix} \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_1 & t_2 & \cdots & t_m \end{bmatrix} b \Longrightarrow \begin{bmatrix} 1 & \mu_t \\ \mu_t & \sigma_t^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} \mu_b \\ \sigma_{tb} \end{bmatrix}.$$

Hence $\alpha_1 = (\mu_b \sigma_t^2 - \mu_t \sigma_{tb})/(\sigma_t^2 - \mu_t^2)$ and $\alpha_2 = (\sigma_{tb} - \mu_t \mu_b)/(\sigma_t^2 - \mu_t^2)$.

The best fit is given by the line $y = \beta(t - \mu_t) + \mu_b$, where $\beta = (\sigma_{tb} - \mu_t \mu_b)/(\sigma_t^2 - \mu_t^2)$.

Linear fit for temperature data

Data for the 5-year temperature averages.

```
year = (1955:5:2000)';
y = [ -0.0480; -0.0180; -0.0360; -0.0120; -0.0040;
       0.1180; 0.2100; 0.3320; 0.3340; 0.4560 ];

t = (year - 1955) / 10;     % better matrix conditioning later
A = [ t.^0 t ];             % coefficient matrix
c = A \ y;
f = @(year) polyval(c(end:-1:1), (year - 1955) / 10);

clf
scatter(year, y), axis tight
xlabel('year'), ylabel('anomaly ({\circ}C)')
hold on
fplot(f, [1955, 2000])
```

Linear fit for temperature data

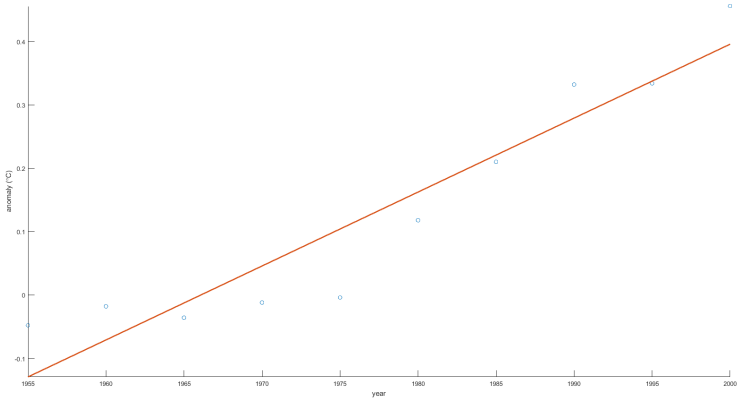


Figure : St-line fit for 5-year average temperature data points.

Polynomial data fitting problem

For $(n-1)$ degree polynomial $p(t) = x_1 + x_2 t + \cdots + x_n t^{n-1}$ fitting the data $(t_1, b_1), \ldots, (t_m, b_m)$, we have $p(t_i) = b_i + r_i$ for $i = 1 : m$. This yields the LSP

$$\begin{bmatrix} 1 & t_1 & \cdots & t_1^{n-1} \\ 1 & t_2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & t_m & \cdots & t_m^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & t_1 & \cdots & t_1^{n-1} \\ 1 & t_2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & t_m & \cdots & t_m^{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \approx \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

In practice, one considers $n = 2, 3, 4$ which correspond to straight-line, quadratic and cubic polynomial fit, respectively.

The matrix in the LSP has full rank. Hence the LSP $Ax \approx b$ has a unique solution which is obtained by solving the normal equation

$$A^\top A x = A^\top b.$$

However, for large n the matrix becomes highly ill-conditioned.

Example

Consider the data points

t	-1.0	-0.5	0.0	0.5	0.1
b	1.0	0.5	0.0	1.5	2.0

For the quadratic polynomial fit, we have the LSP

$$\begin{bmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 0.1 & 1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \approx \begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 1.5 \\ 2.0 \end{bmatrix}.$$

Solving the LSP we have $x = \begin{bmatrix} 0.086 & 0.40 & 1.4 \end{bmatrix}^\top$ which yields the polynomial $p(t) = 0.086 + 0.4t + 1.4t^2$.

Example

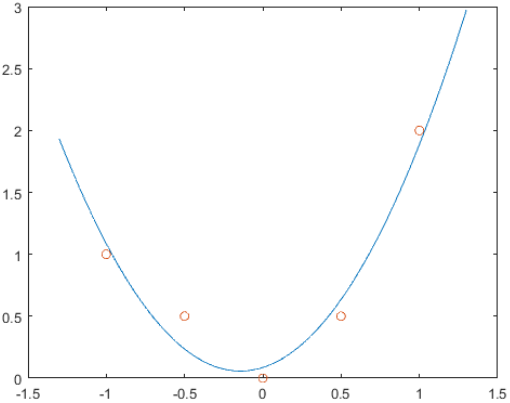


Figure : The plot of $p(t) = 0.086 + 0.4t + 1.4t^2$ and the data points.

Linear and cubic polynomial fit for temperature data

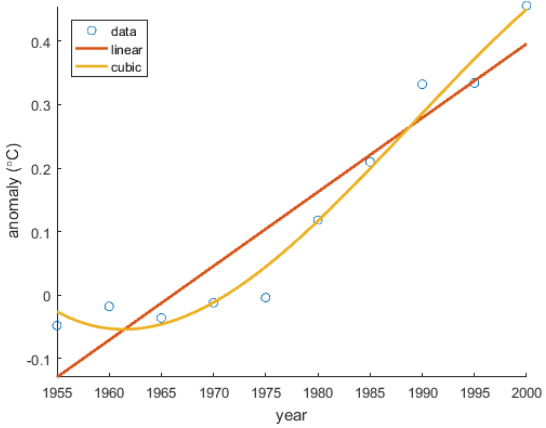


Figure : Linear and cubic fit for 5-year average temperature data points.

Linear parameterized model

Consider the data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ in $\mathbb{R}^n \times \mathbb{R}$ and the regression model given by

$$f(\mathbf{x}, \alpha, v) = \mathbf{x}^\top \alpha + v = \alpha_1 \phi_1(\mathbf{x}) + \cdots + \alpha_n \phi_n(\mathbf{x}) + v$$

where $\phi_j(\mathbf{x}) = \mathbf{e}_j^\top \mathbf{x} = x_j$ for $j = 1 : n$.

- The components of $\mathbf{x}$ are called regressors.
- The regressor model is parameterized by the weight vector α and the intercept v.
- The prediction error $\mathbf{r} \in \mathbb{R}^n$ is given by $r_j = f(\mathbf{x}_j, \alpha, v) - y_j = \mathbf{x}_j^\top \alpha + v - y_j$ for $j = 1 : n$.

The minimization of $\|\mathbf{r}\|_2$ yields the LSP

$$\begin{bmatrix} 1 & \phi_1(\mathbf{x}_1) & \cdots & \phi_n(\mathbf{x}_1) \\ 1 & \phi_1(\mathbf{x}_2) & \cdots & \phi_n(\mathbf{x}_2) \\ \vdots & \vdots & \cdots & \vdots \\ 1 & \phi_1(\mathbf{x}_m) & \cdots & \phi_n(\mathbf{x}_m) \end{bmatrix} \begin{bmatrix} v \\ \alpha \end{bmatrix} = \begin{bmatrix} 1 & \mathbf{x}_1^\top \\ 1 & \mathbf{x}_2^\top \\ \vdots & \vdots \\ 1 & \mathbf{x}_m^\top \end{bmatrix} \begin{bmatrix} v \\ \alpha \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

Least-squares classification

A data fitting problem where the outcome $y = g(\mathbf{x})$ can take only two values 1 and -1 gives a classification algorithm. The values of y represent two categories and $y = g(\mathbf{x})$ is called Boolean classifier.

Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ be training data in $\mathbb{R}^n \times \mathbb{R}$. Determine the regressor model

$$f(\mathbf{x}, \alpha, v) = \mathbf{x}^\top \alpha + v$$

that best fit the data.

Define

$$g(\mathbf{x}) := \operatorname{sign}(f(\mathbf{x}, \alpha, v)) = \begin{cases} +1 & \text{if } f(\mathbf{x}, \alpha, v) \geq 0 \\ -1 & \text{if } f(\mathbf{x}, \alpha, v) < 0 \end{cases}$$

Then $y = g(\mathbf{x})$ is a least-squares Boolean classifier.

Least-squares problem (LSP)

Let $A \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$. Usually $m \gg n$. Find $x \in \mathbb{C}^n$ that minimizes

$$\|Ax - b\|_2^2 = \sum_{i=1}^{m} \Big|\Big(\sum_{j=1}^{n} a_{ij}x_j - b_i\Big)\Big|^2.$$

This is called least-squares problem because we minimize the sum of the squares of the errors

$$|r_1|^2 + \cdots + |r_m|^2 \text{ where } r := Ax - b.$$

The vector $r := Ax - b$ is called residual vector and $\|r\|_2$ is called residual error of the least squares problem. We write a solution x of the LSP as

$$x = \arg\min_{y\in\mathbb{C}^n} \|Ay - b\|_2.$$

The LSP is called a linear least squares problem and is written as

$$\text{solve } Ax \approx b \text{ or } \quad \text{LSP } Ax \approx b.$$

Remark: If x is a solution of the LSP $Ax \approx b$ then so is $x + z$ for any $z \in N(A)$.

Normal equation

If $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$, then define

$$f(x) := \|Ax - b\|_2^2 = \sum_{i=1}^{m} \Big(\sum_{j=1}^{n} a_{ij}x_j - b_i \Big)^2.$$

Then gradient $\nabla f(x) = 2A^\top(Ax - b)$ and Hessian $H_f(x) = A^\top A$. For a minimum $\nabla f(x) = 0$ yields the normal equation $A^\top Ax = A^\top b$. Since Hessian is symmetric positive semidefinite, f has a minimum at x.

Note that $A^\top A$ is positive semidefinite and the normal equation

$$A^\top Ax = A^\top b$$

is always consistent and has a solution. If $\mathrm{rank}(A) = n$ then $A^\top A$ is positive definite and $x = (A^\top A)^{-1}A^\top b = A^+ b$ is a unique solution of the LSP $Ax \approx b$.

Remark: If $\mathrm{rank}(A) = n$ then $A^\top Ax = A^\top b$ can be solved by Cholesky factorization. However, $A^\top A$ may be highly ill-conditioned.

Geometry of Least-squares problem

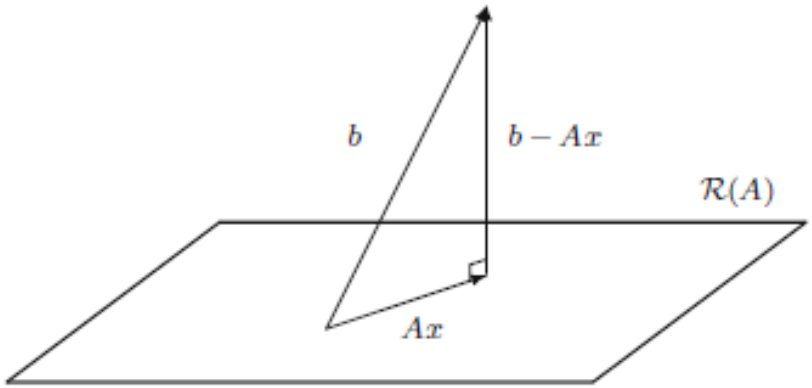


Figure : Relationships among b, r and $R(A)$.

Normal equation

Consider $R(A) := \{Ax : x \in \mathbb{C}^n\} \subset \mathbb{C}^m$ and $N(A) := \{x \in \mathbb{C}^n : Ax = 0\} \subset \mathbb{C}^n$ of $A \in \mathbb{C}^{m\times n}$. Then we have

$$\begin{aligned} \mathbb{C}^m &= R(A) \oplus N(A^*) \text{ and } R(A) \perp N(A^*) \\ \mathbb{C}^n &= N(A) \oplus R(A^*) \text{ and } N(A) \perp R(A^*). \end{aligned}$$

Theorem: The LSP $Ax = b$ has a solution and

$$x = \underset{y\in\mathbb{C}^n}{\operatorname{argmin}} \|Ay - b\|_2 \iff (Ax - b) \perp R(A) \iff A^*Ax = A^*b.$$

Proof: $(Ax - b) \perp R(A) \iff b - Ax \in N(A^*) \iff A^*(Ax - b) = 0 \iff A^*Ax = A^*b.$

Let $b = b_1 + b_2$ with $b_1 \in R(A)$ and $b_2 \in N(A^*)$. Then

$$\begin{aligned} \|Ax - b\|_2^2 &= \|Ax - b_1 - b_2\|_2^2 = \|Ax - b_1\|_2^2 + \|b_2\|_2^2 = \|b_2\|_2^2 \\ &\Leftrightarrow Ax = b_1 \iff b_2 = b - Ax \iff (Ax - b) \perp R(A). \blacksquare \end{aligned}$$

The system $A^*Ax = A^*b$ is called the normal equation for $Ax \approx b$.

Regularized Least-squares problem

Linear measurement model estimation: $y = Ax + v$

- x is a n-vector containing parameter that we wish to estimate
- v is an m-vector containing unknown measurement error or noise
- y is an m-vector containing measurement and A is an $m \times n$ matrix

Regularized LSP (Tikhonov regularization): $\min_x \left(\|Ax - y\|_2^2 + \lambda \|Dx\|_2^2\right)$ where $\lambda > 0$.

The goal is to make both $\|Ax - y\|_2$ and $\|Dx\|_2$ small. The solution is unique when $\mathrm{rank}(D) = n$. Indeed,

$$\|Ax - y\|_2^2 + \lambda \|Dx\|_2^2 = \left\| \begin{bmatrix} A \\ \sqrt{\lambda} D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|_2^2 \Longrightarrow (A^*A + \lambda D^*D)x = A^*y.$$

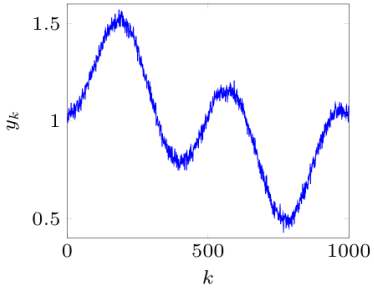
This shows that $x = (A^*A + \lambda D^*D)^{-1} A^*y$ is the unique solution of the regularized LSP.

Signal denoising

- observed signal is n-vector

$$y = x + v$$

- x is unknown signal
- v is noise



Least squares denoising

$$\text{minimize} \quad \|x - y\|^2 + \lambda \sum_{i=1}^{n-1} (x_i - x_{i-1})^2$$

goal is to find slowly varying signal $\hat{x}$, close to observed signal y

Matrix formulation

$$\text{minimize} \quad \left\| \begin{bmatrix} I \\ \sqrt{\lambda} D \end{bmatrix} x - \begin{bmatrix} y \\ 0 \end{bmatrix} \right\|^2$$

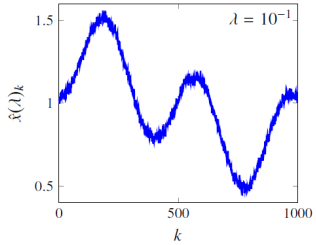
- D is $(n-1) \times n$ finite difference matrix

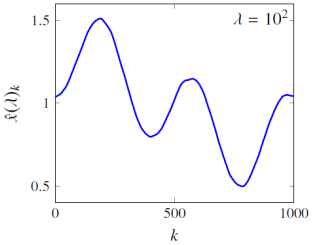
$$D = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix}$$

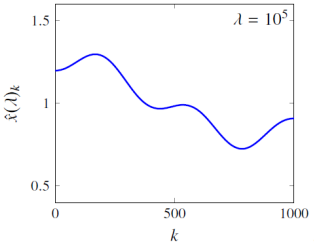
- equivalent to linear equation

$$(I + \lambda D^T D)x = y$$

Three solutions







- $\hat{x}(\lambda) \to y$ for $\lambda \to 0$
- $\hat{x}(\lambda) \to \mathbf{avg}(y)\mathbf{1}$ for $\lambda \to \infty$
- $\lambda \approx 10^2$ is good compromise

MA580H Matrix Computations

Least-Squares Problem(LSP)

Lecture 12: QR method for LSP

Rafikul Alam
Department of Mathematics
IIT Guwahati

Outline

- QR method for LSP
- Augmented QR method for LSP
- Rank revealing QR method for rank deficient LSP

Unitary matrices

Complex matrix	Real matrix
Unitary: $AA^* = A^*A = I$	Orthogonal: $AA^\top = A^\top A = I$
Isometry: $A^*A = I$	Isometry: $A^\top A = I$

Fact: An $n \times n$ matrix A is unitary (resp., orthogonal) if and only if columns of A are orthonormal. An $m \times n$ matrix is isometry if and only if columns of A are orthonormal.

Example: The matrix $U := \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} \\ \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ is unitary and $P := \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$ is orthogonal. The matrix $Q := \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{bmatrix}$ is an isometry.

Let $Q \in \mathbb{R}^{n\times n}$ be orthogonal. Then $\|Qx\|_2 = \sqrt{\langle Qx,\, Qx\rangle} = \sqrt{\langle x,\, Q^\top Qx\rangle} = \sqrt{\langle x,\, x\rangle} = \|x\|_2$.

Let $Q \in \mathbb{C}^{n\times n}$ be unitary. Then $\|Qx\|_2 = \sqrt{\langle Qx,\, Qx\rangle} = \sqrt{\langle x,\, Q^*Qx\rangle} = \sqrt{\langle x,\, x\rangle} = \|x\|_2$.

QR factorization

Theorem: Let $A \in \mathbb{C}^{m\times n}$. Then there is a unitary matrix $Q \in \mathbb{C}^{m\times m}$ and an upper triangular matrix $\mathcal{R} \in \mathbb{C}^{m\times n}$ such that $A = Q\mathcal{R}$. If $\operatorname{rank}(A) = n$ then $\mathcal{R}$ is of the form $\mathcal{R} = \begin{bmatrix} R \\ 0 \end{bmatrix}$ for some nonsingular upper triangular matrix $R \in \mathbb{C}^{n\times n}$. Let $Q = \begin{bmatrix} Q_n & Q_{m-n} \end{bmatrix}$ with $Q_n \in \mathbb{C}^{m\times n}$. Then Q_n is an isometry and

$$A = Q\mathcal{R} = \begin{bmatrix} Q_n & Q_{m-n} \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix} = Q_n R. \ \blacksquare$$

Remark: The factorization $A = Q\mathcal{R}$ is called a full QR factorization and $A = Q_n R$ is called a compact (or economy size) QR factorization of A. MATLAB commands: `[Q,R] = qr(A)` and `[Q,R] = qr(A,0)`, respectively, compute full and compact QR factorization of A

Example:

$$\underbrace{\begin{bmatrix} 1 & 3 & 1 \\ 1 & 3 & 7 \\ 1 & -1 & -4 \\ 1 & -1 & 2 \end{bmatrix}}_{A} = \underbrace{\frac{1}{2}\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} 2 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathcal{R}}.$$

QR method for LSP

A QR factorization of A provides an efficient method for solution of the LSP $Ax \approx b$.

Suppose that $\mathrm{rank}(A) = n$. Set $\begin{bmatrix} c \\ d \end{bmatrix} := Q^* b$, where $c \in \mathbb{C}^n$ and $d \in \mathbb{C}^{m-n}$. Then

$$\|Ax - b\|_2 = \|Q^*(Ax - b)\|_2 = \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} x - \begin{bmatrix} c \\ d \end{bmatrix} \right\|_2 = \sqrt{\|Rx - c\|_2^2 + \|d\|_2^2}.$$

This shows that $\min \|Ax - b\|_2 = \|d\|_2 \iff Rx = c$. Hence $x = R^{-1}c$ is a unique solution of the LSP and $\|d\|_2$ is the residual. [If $Q \in \mathbb{C}^{n \times n}$ is unitary then $\|Qx\|_2 = \|x\|_2$ for all x.]

Algorithm: Solution of LSP $Ax \approx b$ when $\mathrm{rank}(A) = n$.

1. Compute QR factorization $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$.
2. Set $\begin{bmatrix} c \\ d \end{bmatrix} := Q^* b$, where $c \in \mathbb{C}^n$ and $d \in \mathbb{C}^{m-n}$.
3. Solve upper triangular system $Rx = c$.
4. Compute the residual $\|d\|_2$.

Example

Given $A = \begin{bmatrix} 3 & -6 \\ 4 & -8 \\ 0 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} -1 \\ 7 \\ 2 \end{bmatrix}$, solve the LSP $Ax \approx b$.

1. Compute QR factorization: $A = \begin{bmatrix} -\frac{3}{5} & 0 & \frac{4}{5} \\ -\frac{4}{5} & 0 & -\frac{3}{5} \\ 0 & -1 & 0 \end{bmatrix} \left[\begin{array}{cc} -5 & 10 \\ 0 & -1 \\ \hline 0 & 0 \end{array}\right] = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$.
2. Compute $Q^\top b = \left[\begin{array}{c} -5 \\ -2 \\ \hline -5 \end{array}\right]$.
3. Solve $\begin{bmatrix} -5 & 10 \\ 0 & -1 \end{bmatrix} x = \begin{bmatrix} -5 \\ -2 \end{bmatrix} \Longrightarrow x = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$.
4. The residual $\|r\|_2 = 5$.

QR factorization of augmented matrix

If the matrix Q in the QR factorization $A = QR$ is not required, then the LSP $Ax \approx b$ can be solved more efficiently by computing QR factorization of the augmented matrix $\begin{bmatrix} A & b \end{bmatrix}$.

Suppose that $\mathrm{rank}(A) = n$. Then

$$Ax - b = \begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} x \\ -1 \end{bmatrix} \text{ and } \begin{bmatrix} A & b \end{bmatrix} = Q \left[\begin{array}{c|c} R & c \\ 0 & d \\ \hline \multicolumn{2}{c}{\mathbf{0}} \end{array}\right] = Q\mathcal{R}, \text{ where } d \in \mathbb{C}.$$

Hence $\|Ax - b\|_2 = \sqrt{\|Rx - c\|_2 + |d|^2} \Longrightarrow \min \|Ax - b\|_2 = |d| \iff Rx = c.$

Hence the LSP $Ax \approx b$ can be solved in three steps:

- Compute QR factorization `[A, b] = QR`.
- Solve the upper triangular sytem `R(1:n, 1:n)x = R(1:n, n+1)`.
- Compute residual norm `|d| = abs( R(n+1, n+1) )`.

QR method for rank deficient LSP

Theorem: Let $A \in \mathbb{C}^{m\times n}$. Suppose that $\mathrm{rank}(A) = r$. Then there is a unitary matrix $Q \in \mathbb{C}^{m\times m}$ and a nonsingular upper triangular matrix $R_{11} \in \mathbb{C}^{r\times r}$ such that

$$AP = Q\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} = Q\mathcal{R},$$

where $P \in \mathbb{R}^{n\times n}$ is a permutation matrix and $R_{12} \in \mathbb{C}^{r\times(n-r)}$. ■

Set $\begin{bmatrix} c \\ d \end{bmatrix} := Q^*b$, where $c \in \mathbb{C}^r$ and $d \in \mathbb{C}^{m-r}$. Then

$$\begin{aligned} \|Ax - b\|_2 &= \|Q^*(APP^\top x - b)\|_2 = \left\| \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} - \begin{bmatrix} c \\ d \end{bmatrix} \right\|_2 \\ &= \sqrt{\|R_{11}y + R_{12}z - c\|_2^2 + \|d\|_2^2}. \end{aligned}$$

This shows that $\min \|Ax - b\|_2 = \|d\|_2 \iff R_{11}y = c - R_{12}z$. Hence $x = P\begin{bmatrix} y \\ z \end{bmatrix}$ is a solution of the LSP for any $z \in \mathbb{C}^{n-r}$ and $\|d\|_2$ is the residual. Setting $z = 0$ we obtain a unique solution with smallest norm.

QR method for rank deficient LSP

Algorithm: Solution of LSP $Ax \approx b$ when $\mathrm{rank}(A) = r$.

1. Compute QR factorization $AP = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$, where $Q \in \mathbb{C}^{m\times m}$ is unitary, $R_{11} \in \mathbb{C}^{r\times r}$ is nonsingular and upper triangular.
2. Set $\begin{bmatrix} c \\ d \end{bmatrix} := Q^* b$, where $c \in \mathbb{C}^r$ and $d \in \mathbb{C}^{m-r}$.
3. Solve upper triangular system $R_{11} y = c$.
4. Set $x = P \begin{bmatrix} y \\ 0 \end{bmatrix}$. Then x is a unique solution of $Ax \approx b$.
5. Compute the residual $\|d\|_2$.

Remark: If x is a solution of LSP $Ax \approx b$ then $x + z$ is also a solution for any $z \in N(A)$. Hence the LSP has $n - r$ linearly independent solutions.

A rank deficient LSP is an ill-posed problem and solutions are strongly dependent on the rank of A. Numerical rank determination is a tricky problem.