Numerical Linear Algebra

Chenghao Dong

January 1, 2024

CONTENTS

1 Basic LA Review

1.1 Notations and Basic Concepts

在本笔记中，规定如下记号（其他常见记号比如 A^{-1}，$A^T \cdots$ 在此处省略）；此外，未经特殊说明，所有的向量 $\boldsymbol{x} \in \mathbb{R}^n$ 均指代列向量 $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$

1） $\lambda_i(A)$ 表示矩阵 A 按递减排列的第 i 个模最大的特征值。$\lambda(A)$ 表示 A 的所有特征值；

2） $\sigma_i(A)$ 表示第 i 个（最大）的奇异值。$\sigma(A)$ 表示矩阵 A 的所有奇异值；

3） $\mathrm{diag}(A)$ 表示矩阵 A 的对角元素；$\mathrm{span}(A)$ 表示矩阵 A 的列空间；

4） $\overline{A}$ 表示矩阵 A 的共轭阵（conjucate）s.t. $(\overline{A})_{ij} = \overline{A_{ij}}$；

5） A^H 表示矩阵 A 的共轭转置（conjucate transpose）s.t. $A^H = (\overline{A})^T = \overline{A^T}$；

6） $||\boldsymbol{x}||_p$ 表示向量 $\boldsymbol{x}$ 的 p-范数，当 $p = 2$ 时，该范数为常见的欧氏范数（Euclidean norm），简单记作 $||\boldsymbol{x}|| := ||\boldsymbol{x}||_2 = \boldsymbol{x}^H\boldsymbol{x}$；

7） $||A||_p$ 表示矩阵 A 的诱导 p-范数（Induced p-norm），当 $p = 2$ 时，该范数称为矩阵的谱范数（spectral norm），简单记作 $||A|| := ||A||_2$；

8） $||A||_F$ 表示矩阵 A 的诱导 F-范数（Frobenius norm）；

9） $A \succ 0$ 表示矩阵 A 为正定矩阵；$A \succ B$ 表示 $A - B \succ 0$（resp. $\prec$ 表示负定）；$A \succeq 0$ 表示矩阵 A 为半正定矩阵；$A \succeq B$ 表示 $A - B \succeq 0$（resp. $\preceq$ 表示半负定）；

10） $\Lambda/U/L$ 未经特殊说明 Λ 均指代对角矩阵；U 均指代上三角矩阵；L 均指代下三角矩阵；

11） $a_{i.}^T$ / $a_{.j}$ 未经特殊说明 $a_{i.}^T = e_i^T A$ 均指代矩阵 A 的第 i 行；$a_{.j} = Ae_j$ 均指代矩阵 A 的第 j 列，$a_{i.}$，$a_{.j}$ 均视为列向量；所以一个 $m \times n$ 矩阵 A 可以视为：

$$A = \begin{bmatrix} a_{1.}^T \\ \vdots \\ a_{m.}^T \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} a_{.1} & \cdots & a_{.n} \end{bmatrix}$$

12） $\odot$ 表示对应位置的 element-wise 点乘；与一般的矩阵或向量乘区分；

13） $|a|$ 表示 $a \in \mathbb{C}$ 的模 modulus，当 a 为实数时等于绝对值；

14） $\mathbb{F}$ 表示实数域 $\mathbb{R}$ 或复数域 $\mathbb{C}$。

1.2 Matrices Calculations

下面回顾一些常见的**矩阵运算的通式**，主要涉及四类常见的运算：

1） AB 表示矩阵乘法，对于 $A \in \mathbb{F}^{m\times l}$，$B \in \mathbb{F}^{l\times n}$ 其通式

$$(AB)_{ij} = \sum_{k=1}^{l} a_{ik}b_{kj}$$

此外通过将矩阵进行行列拆分，矩阵乘法还可以化为

$$AB = A\begin{bmatrix} b_{\cdot 1} & \cdots & b_{\cdot n}\end{bmatrix} = \begin{bmatrix} Ab_{\cdot 1} & \cdots & Ab_{\cdot n}\end{bmatrix} \text{ s.t. } Ab_{\cdot j} \in \mathbb{F}^m$$

或者

$$AB = \begin{bmatrix} a_{1\cdot}^T \\ \vdots \\ a_{m\cdot}^T \end{bmatrix} B = \begin{bmatrix} a_{1\cdot}^T B \\ \vdots \\ a_{m\cdot}^T B \end{bmatrix} \text{ s.t. } a_{i\cdot}^T B \in \mathbb{F}^{1\times n}$$

这分别说明了 AB 的列是 A 的列的线性组合；行是 B 的行的线性组合，于是显然有 $\operatorname{rank}(AB) \le \min\{\operatorname{rank}(A), \operatorname{rank}(B)\}$；如果 $A = \Lambda \in \mathbb{F}^{l\times l}$ 为对角阵，ΛB 相当于给 B 的行乘上对角阵中对应位置的值

$$\Lambda B = \begin{bmatrix} \Lambda b_{\cdot 1} & \cdots & \Lambda b_{\cdot n}\end{bmatrix} = \begin{bmatrix} \lambda_1 b_{1\cdot}^T \\ \vdots \\ \lambda_l b_{l\cdot}^T \end{bmatrix} \text{ s.t. } \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_l \end{bmatrix}$$

如果 $B = \Lambda \in \mathbb{F}^{l\times l}$ 为对角阵，$A\Lambda$ 相当于给 A 的列乘上对角阵中对应位置的值

$$A\Lambda = \begin{bmatrix} a_{1\cdot}^T \Lambda \\ \vdots \\ a_{m\cdot}^T \Lambda \end{bmatrix} = \begin{bmatrix} \lambda_1 a_{\cdot 1} & \cdots & \lambda_l a_{\cdot l}\end{bmatrix} \text{ s.t. } \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_l \end{bmatrix}$$

2） $\boldsymbol{y}^T A\boldsymbol{x}$ 双线性型（bilinear form）常出现于二次型运算，其结果为一常数，有通式

$$\boldsymbol{y}^T A\boldsymbol{x} = \sum_{ij} y_i a_{ij} x_j$$

3） $X\Lambda Y$ 在 $X \in \mathbb{F}^{m\times l}$，$\Lambda \in \mathbb{F}^{l\times l}$，$Y \in \mathbb{F}^{l\times n}$，其中 Λ 为对角矩阵时，结果是一个 $\mathbb{F}^{m\times n}$ 中的矩阵，且可以分解为X 的列 $\boldsymbol{x}_{\cdot k}$ 与 Y 中对应行 $\boldsymbol{y}_{k\cdot}^T$ 对于相应对角值 λ_k 的加权和

$$X\Lambda Y = \begin{bmatrix} \boldsymbol{x}_{\cdot 1} & \cdots & \boldsymbol{x}_{\cdot l}\end{bmatrix} \Lambda Y = \begin{bmatrix} \boldsymbol{x}_{\cdot 1} & \cdots & \boldsymbol{x}_{\cdot l}\end{bmatrix} \begin{bmatrix} \lambda_1 \boldsymbol{y}_{1\cdot}^T \\ \cdots \\ \lambda_l \boldsymbol{y}_{l\cdot}^T \end{bmatrix} = \sum_{k=1}^{l} \lambda_k \boldsymbol{x}_{\cdot k}\boldsymbol{y}_{k\cdot}^T$$

其中第 (i,j) 个元素等于 $(X\Lambda Y)_{ij} = \sum_{k=l}^{l} \lambda_k x_{ik} y_{kj} = \boldsymbol{x}_{i\cdot}^T \Lambda \boldsymbol{y}_{\cdot j}$

4) XAY 在 $X \in \mathbb{F}^{m\times l}$, $A \in \mathbb{F}^{l\times k}$, $Y \in \mathbb{F}^{k\times n}$ 时，结果是一个 $\mathbb{F}^{m\times n}$ 中的矩阵，且第 (i,j) 个元素等于X 的 i 行 $\boldsymbol{x}_{i.}^T$ 与 Y 中 j 列 $\boldsymbol{y}_{.j}$ 对于矩阵 A 的双线性型：

$$(XAY)_{ij} = \boldsymbol{x}_{i.}^T A \boldsymbol{y}_{.j}$$

一个简单的证明如下

$$XAY = \begin{bmatrix} \boldsymbol{x}_{1.}^T \\ \vdots \\ \boldsymbol{x}_{m.}^T \end{bmatrix} \begin{bmatrix} A\boldsymbol{y}_{.1} & \cdots & A\boldsymbol{y}_{.n} \end{bmatrix} == \begin{bmatrix} \boldsymbol{x}_{1.}^T A\boldsymbol{y}_{.1} & \cdots & \boldsymbol{x}_{1.}^T A\boldsymbol{y}_{.n} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}_{m.}^T A\boldsymbol{y}_{.1} & \cdots & \boldsymbol{x}_{m.}^T A\boldsymbol{y}_{.n} \end{bmatrix}$$

5) $\mathrm{trace}(AB)$ 若 AB 为方阵，则不难得出：$\mathrm{trace}(AB) = \mathrm{trace}(BA)$

6) Neumann series 只要方阵$||X|| < 1$ 对任何一项范数成立，则在该范数下：

$$(I - X)^{-1} = 1 + X + X^2 + X^3 + \cdots$$

7) Matrix Exponentials 定义矩阵指数 e^X s.t. $X \in \mathbb{F}^{n\times n}$ 为方阵，则

$$e^X = 1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \cdots$$

这一般可以通过对角化简化运算。

8) 规范正交化 Othonomalization 如果一个矩阵满足 $A^H A = \Lambda$（Λ 为对角阵且对角线元素不为 0），说明其各列相互正交。此时存在一个对角矩阵 Σ，与一个列规范正交矩阵 $\tilde{A}$ s.t. $\tilde{A}^H \tilde{A} = I$，使得 $A = \tilde{A}\Sigma$，即

$$\begin{bmatrix} a_{.1} & \cdots & a_{.n} \end{bmatrix} = \begin{bmatrix} \frac{a_{.1}}{||a_{.1}||} & \cdots & \frac{a_{.n}}{||a_{.n}||} \end{bmatrix} \begin{bmatrix} ||a_{.1}|| & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & ||a_{.n}|| \end{bmatrix}$$

这对于行正交矩阵同理，只不过变成左乘系数矩阵。

1.3 Matrix Norms

接着我们定义几个**常见的向量与矩阵范数**，从范数的基本定义开始。

Definition 1.3.1. 范数（norms） 线性空间 $(V, \mathbb{F})$ 里的一个范数 $||\cdot||$ 应当满足如下几条性质

$$1) \quad ||\alpha \boldsymbol{x}|| = \alpha ||\boldsymbol{x}||$$

$$2) \quad ||\boldsymbol{x}|| \geq 0 \quad \text{and} \quad ||\boldsymbol{x}|| = 0 \Longleftrightarrow \boldsymbol{x} = \boldsymbol{0}$$

$$3) \quad ||\boldsymbol{x} + \boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$$

Definition 1.3.2. 向量的 p-范数（p-norms） 矢量 $\boldsymbol{x} \in \mathbb{C}^n$ 的 p-范数定义为，对 $1 \leq p \leq \infty$

$$||\boldsymbol{x}||_p = \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}} \quad \text{or} \quad ||\boldsymbol{x}||_\infty = \max_i |x_i|$$

对于复数来说，$|x_i|$ 表示其模长 *modulus*。当 $p = 2$ 时，该范数为常见的欧氏范数（*Euclidean norm*），省略下标简单记作 $||\boldsymbol{x}|| := ||\boldsymbol{x}||_2 = \boldsymbol{x}^T \overline{\boldsymbol{x}}$。

Definition 1.3.3. 矩阵的诱导 p-范数（induced p-norms） 矩阵 A 的诱导 p-范数定义为，对 $1 \leq p \leq \infty$

$$||A||_p = \max_{\boldsymbol{x} \neq \boldsymbol{0}} \frac{||A\boldsymbol{x}||_p}{||\boldsymbol{x}||_p} = \max_{||\boldsymbol{x}||_p = 1} ||A\boldsymbol{x}||_p$$

当 $p = 2$ 时，该范数称为矩阵的谱范数（*spectral norm*），省略下标简单记作 $||A|| := ||A||_2 = \max_{||\boldsymbol{x}||=1} ||A\boldsymbol{x}||$。诱导范数实际上衡量了矩阵在对应向量范数下能对其施加的最大影响。

Definition 1.3.4. 矩阵的 F-范数（F-norms） 矩阵 A 的 F-范数（*Frobenius norm*）定义为

$$||A||_F = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sqrt{\text{trace}(A^H A)}$$

Definition 1.3.5. 矩阵的迹范数/核范数（trace/nuclear-norms） 矩阵 A 的迹范数/核范数定义为

$$||A||_* = \sum_{i=1}^{\min(m,n)} \sigma_i(A) = \max_{Q \ orthonomal} \{\text{trace}(Q^T A)\}$$

Definition 1.3.6. 酉不变范数（unitarily invariant） 向量的一个范数 $||\cdot||$ 是酉不变的当且仅当对于任意酉矩阵 $A^HA = AA^H = I$，其满足 $\forall\, \boldsymbol{x}$: $||A\boldsymbol{x}|| = ||\boldsymbol{x}||$；矩阵的一个范数 $||\cdot||$ 是酉不变的当且仅当对于任意酉矩阵 U, V（注意这里的 U 不是上三角），其满足 $\forall\, A$: $||UAV|| = ||A||$

Theorem 1.3.1. 常见向量范数性质

1） $\frac{1}{n^{1/p}}||\boldsymbol{x}||_p \le ||\boldsymbol{x}||_\infty \le ||\boldsymbol{x}||_p$ 证明仅使用了简单的放缩，略；

2） $\frac{1}{\sqrt{n}}||\boldsymbol{x}||_1 \le ||\boldsymbol{x}||_2 \le ||\boldsymbol{x}||_1$ 前一个不等号使用 *Cauchy-Schwarz*，后一个不等号直接平方加多项式定理展开。

Theorem 1.3.2. 常见矩阵范数性质 对于矩阵范数与 $A \in \mathbb{C}^{m\times n}$:

1） $||AB\boldsymbol{x}||_p \le ||A||_p||B\boldsymbol{x}||_p$（定义）

2） $||AB||_p \le ||A||_p||B||_p$ ； $||AB||_F \le ||A||_F||B||_F$

3） $||A||_1 = \max_j \sum_i |A_{ij}|$ 为最大的列元素模和；

4） $||A||_\infty = \max_i \sum_j |A_{ij}| = ||A^T||_1$ 为最大的行元素模和；

5） $\frac{1}{\sqrt{n}}||A||_\infty \le ||A||_2 \le \sqrt{m}||A||_\infty$ or $\frac{1}{\sqrt{m}}||A||_1 \le ||A||_2 \le \sqrt{n}||A||_1$

6） $||A||_2 \le ||A||_F \le \sqrt{\min(m,n)}||A||_2$

7） $||A||_2 = ||A^T||_2$ ； $||A||_F = ||A^T||_F$

Theorem 1.3.3. 常见范数的酉不变性
向量的 $||\boldsymbol{x}||_2$，矩阵的$||A||_2$，$||A||_F$，$||A||_*$均为酉不变范数；特别地，$||W\boldsymbol{x}||_2 = ||\boldsymbol{x}||_2$ 对列规范正交矩阵 W s.t. $W^HW = I$ 也同样成立。

我们简单证明几个结论：

Proof Theorem 1.2.2.（2） 对于第一个不等式有

$$\forall\, ||\boldsymbol{x}||_p = 1 \ : \ ||AB\boldsymbol{x}||_p \le ||A||_p||B\boldsymbol{x}||_p \le ||A||_p||B||_p$$

由最大值或上确界的定义：$||AB||_p = \sup_{||\boldsymbol{x}||_p=1}||AB\boldsymbol{x}||_p \le ||A||_p||B||_p$

对于第二个不等式有

$$\begin{aligned} ||AB||_F^2 = \textstyle\sum_{i,j} |a_{i.}^T b_{.j}|^2 &\le \textstyle\sum_{i,j} ||a_{i.}||^2 \cdot ||b_{.j}||^2 \\ &= \textstyle\sum_i ||a_{i.}||^2 \cdot \sum_j ||b_{.j}||^2 = ||A||_F^2||B||_F^2 \end{aligned}$$

(3) 记 $\gamma = \text{argmax}_j \sum_i |A_{ij}|$，$\Gamma = \max_j \sum_i |A_{ij}|$，我们先证：$\forall\ ||\boldsymbol{x}||_1 = 1$ ：$||A\boldsymbol{x}||_1 \leq \Gamma$，即

$$\begin{aligned} ||A\boldsymbol{x}||_1 &= \sum_i |\sum_j A_{ij}x_j||_1 = \sum_i |\sum_j A_{ij}x_j| \\ &\leq \sum_i \sum_j |A_{ij}||x_j| = \sum_j |x_j| \sum_i |A_{ij}| \\ &\leq \sum_j |x_j| \cdot \Gamma = \Gamma \sum_j |x_j| = \Gamma ||\boldsymbol{x}||_1 = \Gamma \end{aligned}$$

于是 $||A||_1 = \max\limits_{||\boldsymbol{x}||_1=1} ||A\boldsymbol{x}||_1 \leq \Gamma$，而显然取 $\boldsymbol{x} = e_\gamma$ 时等号 $||A\boldsymbol{x}||_1 = \Gamma$ 成立，故

$$||A||_1 = \Gamma = \max_j \sum_i |A_{ij}|$$

注意 (4) 的证明与之几乎完全类似，故省略。

(5) 只证明前一个不等式。不妨记：$\gamma = \text{argmax}_{||\boldsymbol{x}||_2=1} ||A\boldsymbol{x}||_2$，由 thm 1.2.1. (1) 知道：对于任意的 $||\boldsymbol{x}||_2 = 1$，$A\boldsymbol{x} \in \mathbb{C}^m$ 总满足

$$||A\boldsymbol{x}||_\infty \leq ||A\boldsymbol{x}||_2 \leq \sqrt{m}||A\boldsymbol{x}||_\infty$$

于是

$$||A||_2 = ||A\gamma||_2 \leq \sqrt{m}||A\gamma||_\infty \leq \sqrt{m}||A||_\infty$$

左侧不等号同理。注意第二个不等式的证明与之几乎完全类似，故省略。

(6) 先证左侧不等号：对任意 $||\boldsymbol{x}||_2 = \sqrt{\sum_j^n |x_j|^2} = 1$：

$$\begin{aligned} ||A||_F^2 &= \sum_i^m \sum_j^n |a_{ij}|^2 = \sum_i^m \left(\sum_j^n |a_{ij}|^2\right) \cdot 1 = \sum_i^m \left(\sum_j^n |a_{ij}|^2\right) \cdot ||\boldsymbol{x}||_2^2 \\ &= \sum_i^m \left\{ \left(\sum_j^n |a_{ij}|^2\right) \left(\sum_j^n |x_j|^2\right) \right\} \geq \sum_i^m \left\{ |\sum_j^n a_{ij}x_j|^2 \right\} \quad \text{Cauchy} - \text{Schwartz} \\ &= \sum_i^m |\,(A\boldsymbol{x})_i\,|^2 = ||A\boldsymbol{x}||_2^2 \Longrightarrow ||A||_F \geq \sup_{||\boldsymbol{x}||_2=1} ||A\boldsymbol{x}||_2 = ||A||_2 \end{aligned}$$

再证右侧不等号：显然 $||A||_2^2 = \max_{||\boldsymbol{x}||_2=1} ||A\boldsymbol{x}||_2 \geq ||Ae_j||_2$ for $j = 1 \cdots n$，于是

$$n||A||_2^2 \geq \sum_{j=1}^n ||Ae_j||_2 = ||A||_F$$

由 (7) 知道还有 $m||A^T||_2 = m||A|| \geq ||A||_F$，合并得到 $||A||_F \leq \min(m, n)\, ||A||_2$

(7) 对于 F-norm 显然因为其只不过是将矩阵的所有元素一维化后使用向量的 2-norm；对于 $||A||_2 = ||A^T||_2$ 的证明见奇异值分解 SVD 章节。

Proof Theorem 1.2.3. 先证 $||\boldsymbol{x}||_2$，余下的留待在 SVD 后证明。令 U 为 Unitary：

$$||U\boldsymbol{x}||_2 = (U\boldsymbol{x})^T(U\boldsymbol{x}) = \boldsymbol{x}^T(U^TU)\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{x} = ||\boldsymbol{x}||_2$$

1.4 Structured Matrices and Common Theories

下面是几种常见的**方阵** $A \in \mathbb{C}^{n\times n}$结构和与和它们有关的常见结论。

1) 正规矩阵（normal） 若方阵满足$A^HA = AA^H$则称其为正规矩阵，一个方阵正规当且仅当其在复数域 $\mathbb{C}$可以酉对角化，即$\exists\ U$ whose columns consist of eigenvectors in $\mathbb{C}^n$ s.t. $U^HU = UU^H = I$ unitary and $A = U\Lambda U^H$；或等价地，存在由 n 个特征向量（from $\mathbb{C}^n$）构成的规范正交基底使其张成为 $\mathbb{C}^n$。

2) 酉矩阵（unitary） 若方阵满足$A^HA = AA^H = I$则称其为酉矩阵；若为实数阵，则 $A^TA = AA^T = I$ 称为规范正交矩阵（orthogonal，注意和列正交 have orthogonal columns $A^HA = \Lambda$ 区别）；一个正规矩阵为酉矩阵当且仅当其所有特征值的模均为 1，即 $|\lambda| \equiv 1$。注意正交矩阵的特征值也不一定是实特征值。

3) 自伴/厄米矩阵（self-adjoint/Hermitian） 若复数方阵满足 $A^H = A$ 则称其为自伴/厄米矩阵；若为实数阵，则 $A^T = A$ 称为实对称矩阵（real symmetry）；一个正规矩阵为厄米矩阵当且仅当其所有特征值为实数，即 $\lambda \in \mathbb{R}$。

4) 斜厄米矩阵（skew-Hermitian）若复数方阵满足 $A^H = -A$ 则称其为斜厄米矩阵；若为实数阵，则 $A^T = -A$ 称为斜对称矩阵（skew-symmetry）。

5) 三角方阵（triangular） 三角方阵的结构在同类乘法，取逆（若存在）变换下保持不变。

下面是两种常见的**非方阵** $A \in \mathbb{C}^{m\times n}$结构。

1) Hessenberg 若矩阵满足$A_{ij} = 0 \quad \text{if} \quad i > j + 1$则称其为 Hessenberg 矩阵（下辅对角线以下都为 0）。

2) 列规范正交矩阵（othonormal） 若矩阵满足 $A^HA = I \quad \text{if} \quad m > n$ (tall) 则称其为列规范正交矩阵，或称其为 matrix with orthonormal columns；注意此时 $A^HA = I \nRightarrow AA^H = I$。

3) 稀疏矩阵（sparse） 若矩阵中的大量元素为 0，则称其为稀疏矩阵，否则为 dense。用$\text{nnz}(A)$表示矩阵中非 0 元素的个数。

4) 高/宽矩阵（tall/fat） 若矩阵满足行多于列 $m > n$，则称起为高矩阵，反之 $(m \neq n)$ 称为宽矩阵。

由于**实对称**矩阵有很多很好的性质，我们取出来单独讨论。在此之前先做一些定义：

Definition 1.4.1. 正/负定(positive/negative definite) n 阶实方阵 $A \succ 0$ ($A \prec 0$) 为正/负定的，是指其二次型 $\forall \boldsymbol{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$ ： $\boldsymbol{x}^T A \boldsymbol{x} > 0$ (< 0)；

半正/负定 (positive/negative semi-definite) n 阶实方阵 $A \succeq 0$ ($A \preceq 0$) 为半正/负定的，是指其二次型 $\forall \boldsymbol{x} \in \mathbb{R}^n$ ： $\boldsymbol{x}^T A \boldsymbol{x} \geq 0$ (≤ 0)。

Theorem 1.4.1. 实对称矩阵的性质 若 $A^T = A \in \mathbb{R}^{n \times n}$ 为实对称矩阵，则

1) 特征值分解 所有实对称矩阵都能进行正交对角化：$A = V \Lambda V^T$，其中$VV^T = V^T V = I$为正交矩阵。

2) 特征值和特征向量 所有实对称矩阵的特征值均为实数，特征向量也均来自且能张成实数空间 $\mathbb{R}^n$ (由特征向量的求法自然可得)。

3) 正/负定性质 所有实对称矩阵，下列命题等价：

$$
\begin{aligned}
A \succ 0 &\iff \forall\ \lambda_i \ :\ \lambda_i > 0 \iff \forall\ d_i \ :\ d_i > 0 \\
A \prec 0 &\iff \forall\ \lambda_i \ :\ \lambda_i < 0 \iff \forall\ d_i \ :\ (-1)^i d_i > 0 \\
A \succeq 0 &\iff \forall\ \lambda_i \ :\ \lambda_i \geq 0 \iff \forall\ M_i \ :\ M_i \geq 0 \\
A \preceq 0 &\iff \forall\ \lambda_i \ :\ \lambda_i \leq 0 \iff \forall\ M_i \ :\ (-1)^i M_i \geq 0
\end{aligned}
$$

其中 d_i 与 M_i 分别表示矩阵 A 的顺序主子式 *leading principle minor* 和主子式 *principle minor*。d_i 为前 i 行 i 列分块子矩阵的行列式；M_i 为所有任取 i 行与相同下标的 i 列合并成的子矩阵的行列式。

4) 实对称矩阵的转化 对任何 $B \in \mathbb{C}^{m \times n}$ ，都有 $B^H B$ 与 BB^H 实对称。

1.5 Subspaces and Dimensions

Definition 1.5.1. 向量张成与张成空间的维度
假设有一组列向量 $\{v_i\}_{i=1}^n \subseteq \mathbb{C}^m$，我们称其中极大线性无关组的大小 d 为这些向量张成空间 $\text{span}\{v_i\}_{i=1}^n$ 的维数，这些构成极大线性无关组的向量为张成空间的基底 *basis*。将这些向量依次排开成一个矩阵 $V = [v_{\cdot 1} \ \cdots \ v_{\cdot n}]$，则可等价地记这些向量的张成为 $\text{span}(V)$，这些向量的任何线性组合都可以有等价的矩阵表示：

$$V\boldsymbol{c} = \sum_{i=1}^{n} c_i v_i$$

其中 $\boldsymbol{c} = [c_1 \ \cdots \ c_n]^T$ 为系数向量。不难由定义得到维数 $d = \text{rank}(V)$。

下面是一些线性代数中关于维度的常见定理（所在数域 $\mathbb{C}$, $\mathbb{R}$ 不影响结论）：

Theorem 1.5.1. 向量张成与张成空间的维度 假设有两组列向量（构成的矩阵）$V_1 \in \mathbb{C}^{n\times d_1}$ 与 $V_2 \in \mathbb{C}^{n\times d_2}$ 使得 $d_1 + d_2 > n$，则两个张成空间 $\text{span}(V_1)$ 与 $\text{span}(V_2)$ 必有非 $\mathbf{0}$ 的重合元素：

$$\big(\text{span}(V_1) \cap \text{span}(V_2)\big) \setminus \{\mathbf{0}\} \neq \emptyset$$

证明见 *Lecture Notes p.13*。

Theorem 1.5.2. 维度定理（dimension theorem） 假设有一个矩阵 $V \in \mathbb{C}^{m\times n}$：$\mathbb{C}^n \mapsto \mathbb{C}^m$，则：

$$\text{nullity}(V) + \text{rank}(V) = n$$

回顾一下：$\text{nullity}(V) = \dim(N(V))$，其中 $N(V) = \{\boldsymbol{x} \in \mathbb{C}^n : V\boldsymbol{x} = \mathbf{0}\}$ 称为矩阵的零空间 *null space*；$\text{rank}(V) = \dim(R(V))$，其中 $R(V) = \text{span}(V) \subseteq \mathbb{C}^m$ 称为矩阵的列空间 *column space*，或者值域 *range*。证明见高等代数笔记。

关于特征多项式、特征值还有特征空间还有几个常见的结论如下：

Theorem 1.5.3. 特征多项式/特征值/特征空间的性质
记 $A \in \mathbb{C}^{n\times n}$ 为方阵，其有$k \leq n$个相异的特征值，特征多项式为

$$f(\lambda) = \det(\lambda I - A) = \prod_{i=1}^{k} (\lambda - \lambda_i)^{m_i}$$

我们称 m_i 为特征值对应的代数重数 *algebraic multiplicity*；$\dim(E_{\lambda_i}) = \dim(A - \lambda_i I)$ 为特征值对应的几何重数 *geometric multiplicity*，我们有如下结论：

1) 行列式与迹的计算 有行列式计算公式 $\det(A)=\prod_{i=1}^{k}\lambda_i^{m_i}$ 与迹的计算公式 $\operatorname{trace}(A)=\sum_{i=1}^{k}m_i\lambda_i$（同时对 $f(\lambda)$ 的两种形式展开即可）。

2) 特征值与可逆 A 可逆 $\iff$ 特征值均不为 0 $\iff$ 奇异值均不为 0 $\iff$ $A^HA\succ 0$。第一条：存在 0 特征值当且仅当 $A\boldsymbol{x}=0$ 有非平凡解；第二、三条详见后文 *thm 2.1.1* 与 *def 2.1.1*，非 0 奇异值的个数等于矩阵的秩。

3) A^H 的特征值 若 λ_i 为 A 的特征值，则 $\overline{\lambda}_i$ 必为 A^H 的特征值。一个简单的证明如下：

$$f_{A^H}(\overline{\lambda_i})=\det(\overline{\lambda_i}I-A^H)=\det(\overline{\lambda_i}I-\overline{A})=\overline{\det(\lambda_iI-A)}=\overline{f_A(\lambda_i)}=0$$

4) 重数与对角化 几何重数不超过代数重数 $1\le\dim(E_{\lambda_i})\le m_i$，且当且仅当 $\forall\,\lambda_i:\dim(E_{\lambda_i})=m_i$ 时 A 才可以对角化。

5) 相似矩阵的特征值 相似矩阵 A，B s.t. $A=PBP^{-1}$，有相同的特征多项式（因此特征值、迹、行列式均相同，秩也相同）。

6) *Cayley Hamilton Theorem* 特征多项式是矩阵的零化多项式 $f(A)=0$。

Lemma 1.5.1. 设 $A\in\mathbb{C}^{m\times n}$，则

$$\begin{bmatrix}I_m & A_{mn}\\ O & I_n\end{bmatrix}^{-1}=\begin{bmatrix}I_m & -A_{mn}\\ O & I_n\end{bmatrix}$$

Theorem 1.5.4. 设 $A\in\mathbb{C}^{m\times n}$，$B\in\mathbb{C}^{n\times m}$，则 AB 与 BA 有相同的特征值。特征多项式关系：$f_{AB}(\lambda)=\lambda^{m-n}f_{BA}(\lambda)$。

Proof Theorem 1.5.4. 观察发现：

$$\begin{bmatrix}I_m & -A_{mn}\\ O & I_n\end{bmatrix}\begin{bmatrix}AB & O\\ B_{nm} & O_n\end{bmatrix}\begin{bmatrix}I_m & A_{mn}\\ O & I_n\end{bmatrix}=\begin{bmatrix}O_m & O\\ B & BA\end{bmatrix}$$

也即矩阵 $C_1=\begin{bmatrix}AB & O\\ B_{nm} & O_n\end{bmatrix}$ 与 $C_2=\begin{bmatrix}O_m & O\\ B & BA\end{bmatrix}$ 相似且为分块三角矩阵，故有相同的特征多项式。由分块三角矩阵行列式公式，他们的特征多项式可以化为：

$$\left.\begin{aligned}f_{C_1}(\lambda)=|\lambda I-C_1|=|\lambda I-AB|\lambda^n=\lambda^n f_{AB}(\lambda)\\ f_{C_2}(\lambda)=|\lambda I-C_2|=|\lambda I-BA|\lambda^m=\lambda^m f_{BA}(\lambda)\\ f_{C_1}(\lambda)=f_{C_2}(\lambda)\end{aligned}\right\}\Rightarrow f_{AB}(\lambda)=\lambda^{m-n}f_{BA}(\lambda)$$

2 Singular Value Decomposition

2.1 Singular Values

在本章节我们讨论奇异值的定义合法性与一些前置的理论知识及证明。

Theorem 2.1.1. 设 $A \in \mathbb{C}^{m\times n}$，则 A^HA 与 AA^H 均为实对称且：

1） A^HA 与 AA^H 均有实特征根

2） A^HA 与 AA^H 的特征根均非负 $\lambda \geq 0$，即均为半正定矩阵

3） A，A^HA，AA^H 的秩相同，即 $\text{rank}(A) = \text{rank}(A^HA) = \text{rank}(AA^H)$

4） A^HA，AA^H 有相同的非 0 特征值

我们先证明结论 2/3，结论 1 与由结论 4 分别由实对称的性质与 thm 1.5.4 显然可得。

Proof Theorem 2.1.1. **(2)** 假设 A^HA 存在特征根 $\lambda \neq 0$，与对应的特征向量 $\boldsymbol{x}_\lambda$，则考虑 $||A\boldsymbol{x}_\lambda||^2$，有

$$0 \leq ||A\boldsymbol{x}_\lambda||^2 = (A\boldsymbol{x}_\lambda)^H A\boldsymbol{x}_\lambda = \boldsymbol{x}_\lambda^H \left(A^HA\right)\boldsymbol{x}_\lambda = \lambda||\boldsymbol{x}_\lambda^H||^2$$

故有：$\lambda||\boldsymbol{x}_\lambda^H||^2 \geq ||A\boldsymbol{x}_\lambda||^2 \Rightarrow \lambda \geq 0$

(3) 先证明前一个等号：只要证明 $\text{nullity}(A) = \text{nullity}(A^HA)$ 即可由 dimension theorem 自然得到等式结果，实际上：$N(A) = N(A^HA)$。显然 $N(A) \subseteq N(A^HA)$，我们的目标是证明：$N(A^HA) \subseteq N(A)$。现任取：$\boldsymbol{x} \in N(A^HA)$，

$$||A\boldsymbol{x}||^2 = \boldsymbol{x}^H(A^HA\boldsymbol{x}) = \boldsymbol{x}^H\boldsymbol{0} = 0 \Rightarrow ||A\boldsymbol{x}|| = 0 \iff A\boldsymbol{x} = 0$$

即 $\boldsymbol{x} \in N(A)$，所以 $\text{rank}(A) = \text{rank}(A^HA)$，同理

$$\text{rank}(AA^H) = \text{rank}(A^H) = \text{rank}(A)$$

Definition 2.1.1. 矩阵的奇异值（singular values） 任给一个矩阵 $A \in \mathbb{C}_r^{m\times n}$ s.t. $\text{rank}(A) = r$，记 $\lambda_i : i = 1\cdots r$ 为 A^HA 按照降序排列的特征值。定义 A 的奇异值为：

$$\sigma_i(A) = \sqrt{\lambda_i} \quad \text{s.t.} \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$$

上述的定理 *2.1.1* 中的 *1* 与 *2* 确保了开根号的合理性，*3* 确保了非 *0* 奇异值的个数等于 A^HA 的秩（相似对角化），从而又等于原矩阵的秩。

2.2 Singular Value Decomposition

本节中我们将要讨论一种最常见的矩阵分解，它可以将任意形状的实、复数矩阵分解成一个对角矩阵与两个列正交或酉矩阵的乘积，称为奇异值分解 SVD。奇异值分解有很多不同的形式，一种形式如下：

Theorem 2.2.1. 奇异值分解 SVD
任何 $A \in \mathbb{C}^{m\times n}$ s.t. $m \geq n$ *tall*，都可以分解成如下的形式（*thin SVD*）

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

其中 $U^HU = V^HV = I_n$ 为列规范正交矩阵；$\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_n)$ s.t. $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$。注意这几个矩阵的维数：$U \in \mathbb{R}^{m\times n}$，$\Sigma \in \mathbb{R}^{n\times n}$，$V \in \mathbb{R}^{n\times n}$ 称 U，V 的列分别为左、右奇异向量 *singular vectors*。这种方法中 U 不一定是方阵；

或者可以分解成如下形式（*full SVD*），记任何 $A \in \mathbb{C}^{m\times n}$ s.t. $\mathrm{rank}(A) = r$，不一定为 *tall matrix*

$$A = U\Sigma V^H = U \begin{bmatrix} \Sigma_r & O \\ O & O \end{bmatrix} V^H$$

其中 $U^HU = I_m$，$V^HV = I_n$ 为酉方阵；$\Sigma_r = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$ s.t. $\sigma_1 \geq \cdots \geq \sigma_r > 0$。矩阵的维数：$U \in \mathbb{R}^{m\times m}$，$\Sigma \in \mathbb{R}^{m\times n}$，$V \in \mathbb{R}^{n\times n}$ 称 U，V 的列分别为左、右奇异向量。这种方法中 Σ 不一定是方阵

对于 *full SVD* 更详细地，有：

$$A = U\Sigma V^H = \begin{bmatrix} \widetilde{U}_r & \widetilde{U}^{\perp}_{m-r} \end{bmatrix} \begin{bmatrix} \Sigma_r & O_1 \\ O_2 & O_3 \end{bmatrix}_{m\times n} V^H$$

其中 $A^HA = V\Lambda V^H$ 为 A^HA 的正交对角化；$\widetilde{U}_r$ 为 AV 前 r 列正规化后按序合并成的子矩阵（除以 *norm*，*i.e.*, 对应的奇异值 $\sigma_i > 0$）得到；最后 $\widetilde{U}^{\perp}_{m-r}$ 为 $\mathrm{span}(\widetilde{U}_r)$ 在 $\mathbb{C}^m$ 中的规范正交补，不妨取 $\mathrm{span}(\widetilde{U}^{\perp}_{m-r}) = N(\widetilde{U}^H_r)$：

$$AV = U\Sigma = \begin{bmatrix} \widetilde{U}_r\Sigma_r & O_{n-r} \end{bmatrix} = \begin{bmatrix} \widetilde{U}_r & \widetilde{U}^{\perp}_{m-r} \end{bmatrix} \begin{bmatrix} \Sigma_r & O_1 \\ O_2 & O_3 \end{bmatrix}_{m\times n}$$

Theorem 2.2.2. 列空间的正交补 任何 $A \in \mathbb{C}^{m\times n}$ 都有 $\{R(A)\}^{\perp} = N(A^H)$

Proof Theorem 2.2.1. 对任意的 $A \in \mathbb{C}^{m\times n}$ s.t. $\text{rank}(A) = r$，都存在对实对称矩阵 $A^HA \in \mathbb{C}^{n\times n}$ 的正交对角化（特征值从大到小）$A^HA = V\Lambda V^H$，不难证明 AV 为部分列正交矩阵（可能含有 $\mathbf{0}$ 向量）：

$$(AV)^H(AV) = V^H(A^HA)V = V^H(V\Lambda V^H)V = \Lambda$$

于是，我们可以对 AV 进行分解，考虑由相似 $A \sim \Lambda$，可以有

$$\text{rank}(A) = \text{rank}(\Lambda) = \text{number of positive eigenvalues} = r$$

所以 AV 的前 r 个列向量 $u_{\cdot i}$ s.t. $1 \le i \le r$ 不为 0，且其范数就是奇异值 $||u_{\cdot i}|| = \sigma_i$，即有

$$\begin{aligned} AV &= \begin{bmatrix} u_{\cdot 1} & \cdots & u_{\cdot r} & 0 & \cdots & 0 \end{bmatrix} \\ &= \begin{bmatrix} \widetilde{U}_r\Sigma_r & O_{m\times(n-r)} \end{bmatrix} \\ &= \begin{bmatrix} \widetilde{U}_r & \widetilde{U}^{\perp}_{m\times(m-r)} \end{bmatrix} \begin{bmatrix} \Sigma_r & O_{r\times(n-r)} \\ O_{(m-r)\times r} & O_{(m-r)\times(n-r)} \end{bmatrix} \qquad (*) \\ &= U\Sigma \end{aligned}$$

注意由于 $\dim(\text{span}(\widetilde{U}_r)) = r \le m = \dim(\mathbb{C}^m)$，必然能找到 $\text{span}(\widetilde{U}_r)$ 的一个 $(m-r)$ 维正交补，其上任何一组大小为 $(m-r)$ 的线性无关向量均可以化为一组规范正交基底，作为列向量排好成 $\widetilde{U}^{\perp}_{m\times(m-r)}$ 即可。对于满秩矩阵 $\text{rank}(A) = r = \min\{m, n\}$，这一步甚至可以省略。

特别地，我们总可以找 $N(\widetilde{U}_r^H) = \{\text{span}(\widetilde{U}_r)\}^{\perp}$，即列空间的正交补为共轭转置的零空间。最后两边同时乘上 V^H 即可得到分解结果 $A = U\Sigma V^H$

Proof Theorem 2.2.2.

证明一般情况 $\{R(A)\}^{\perp} = N(A^H)$：假设 $\boldsymbol{x} \in R(A)$，$\boldsymbol{y} \in N(A^H)$，则

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = (A\boldsymbol{c})^H\boldsymbol{y} = \boldsymbol{c}^H(A^H\boldsymbol{y}) = \boldsymbol{c}^H \cdot \mathbf{0} = \mathbf{0}$$

下面介绍 SVD 的具体操作流程。

SVD 的具体操作流程

STEP 1 对 A^HA 进行正交对角化，特征值从大到小排列 $A^HA = V\Lambda V^H$ 这里回顾一下 Gram-Schmidt 正交化的流程，假设要对向量组 $\{\boldsymbol{x}_i\}_{i=1}^k$ 进行规范正交化 $\{\boldsymbol{x}_i\}_{i=1}^k \mapsto \{\boldsymbol{y}_i\}_{i=1}^k$，记 $\boldsymbol{x}^1 = \frac{\boldsymbol{x}}{||\boldsymbol{x}||}$ 为一个向量的单位方向向量，则按照如下两步迭代即可：

$$\boldsymbol{y}_1 = \boldsymbol{x}_1^1 \Rightarrow \boldsymbol{y}_{i+1} = \left(\boldsymbol{x}_{i+1} - \sum_{k=1}^{i}\langle \boldsymbol{x}_{i+1}, \boldsymbol{y}_k\rangle \cdot \boldsymbol{y}_k\right)^1$$

STEP 2 计算 AV 并将其中的非零列向量规范正交化，系数作为奇异值矩阵提出 Σ_r，视情况而定是否需要添加规范正交补列向量（AV 无零列向量或 A 行满秩都不需要）：

$$\begin{aligned}
AV &= \begin{bmatrix} u_{\cdot 1} & \cdots & u_{\cdot r} & 0 & \cdots & 0 \end{bmatrix} \\
&= \begin{bmatrix} \widetilde{U}_r\Sigma_r & O_{m\times(n-r)} \end{bmatrix} \xrightarrow{n=r} \widetilde{U}_r\Sigma_r \\
&= \begin{bmatrix} \widetilde{U}_r & \widetilde{U}^{\perp}_{m\times(m-r)} \end{bmatrix} \begin{bmatrix} \Sigma_r & O_{r\times(n-r)} \\ O_{(m-r)\times r} & O_{(m-r)\times(n-r)} \end{bmatrix} \xrightarrow{m=r} \widetilde{U}_r \begin{bmatrix} \Sigma_r & O_{r\times(n-r)} \end{bmatrix} \\
&= U\Sigma
\end{aligned}$$

如果需要找 $\widetilde{U}_r$ 的正交补 $\widetilde{U}^{\perp}_{m\times(m-r)}$，则解方程组：$\widetilde{U}_r^H\boldsymbol{x} = 0$ 并取解空间的规范正交基底即可。

STEP 3 回乘 V^H：$A = U\Sigma V^H$

Theorem 2.2.3. SVD 的应用

设 $A \in \mathbb{C}^{m\times n}$，其秩 $\mathrm{rank}(A) = r$ 对其进行 *full SVD* 使得 $A = U\Sigma V^H$，则有下面的结论。证明比较简单，省略：

1) $A = U\Sigma V^H = \sum_{i=1}^{r}\sigma_i u_{\cdot i}v_{i\cdot}^H$

2) $\mathrm{span}(A) = \mathrm{span}(\,[u_{\cdot 1} \ \cdots \ u_{\cdot r}]\,)$；$\mathrm{span}(A^H) = \mathrm{span}(\,[v_{\cdot 1} \ \cdots \ v_{\cdot r}]\,)$

3) $N(A) = \mathrm{span}(\,\begin{bmatrix} v_{\cdot(r+1)} & \cdots & v_{\cdot n} \end{bmatrix}\,)$

4) $Av_{\cdot i} = \sigma_i u_{\cdot i}$ 且 $u_{\cdot i}^H A = \sigma_i v_{\cdot i}$，$(\sigma_i, u_{\cdot i}, v_{\cdot i})$ 称为 i_{th} *singular triplet*

5) V 是 A^HA 的特征向量矩阵；U 的非零部分 $\widetilde{U}_r$ 是 AA^H 的非零特征值的特征向量矩阵。

2.3 Low-rank Approximation via Truncated SVD

Theorem 2.3.1. 矩阵范数与奇异值 设 $A \in \mathbb{C}^{m\times n}$，其秩 $\text{rank}(A) = r$，则

1) $||A|| = \max_{||\boldsymbol{x}||=1} ||A\boldsymbol{x}|| = \sigma_1(A)$，特别地，若 $A^HA = AA^H$ *normal*，$\sigma_1(A) = \max_i |\lambda_i| = \rho(A)$，$|\cdot|$ 表示 *modulus*，$\rho(A)$ 为谱半径。

2) $||A^{-1}||^{-1} = \min_{||\boldsymbol{x}||=1} ||A\boldsymbol{x}|| = \sigma_r(A)$

3) $||A||_F = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sqrt{\sum_{i=1}^r \sigma_i^2}$

4) $||A||_* = \max\limits_{Q\ orthonomal} \{\text{trace}(Q^TA)\} = \sum_{i=1}^r \sigma_i$

5) 条件数 *condition number* $\kappa(A) := ||A|| \cdot ||A^{-1}|| = \sigma_1(A)/\sigma_r(A)$

现在，对 $A \in \mathbb{C}^{m\times n}$，$\text{rank}(A) = r$ 我们考虑找一个秩为 $k \le r$ 的矩阵 A_k 使得其可以尽可能多地保留 A 的特征，即使得：$A_k = \text{argmin}_{\text{rank}(A')\le k} ||A - A'||$ 我们有如下定理

Theorem 2.3.2. Truncated SVD Approximation Theorem
对 $A \in \mathbb{C}^{m\times n}$，其秩 $\text{rank}(A) = r$，考虑其 *full SVD* 展开，$A = U\Sigma V^H$，记

$$A_k = U_k\Sigma_k V_k^H = \sum_{i=1}^k \sigma_i u_{\cdot i} v_{i\cdot}^H$$

其中 U_k，V_k 分别只包含了前 k 个对应的奇异向量，中间的 Σ_k 为 $k\times k$ 方阵，则对 $\forall k \le r$

$$||A - A_k|| = \min_{\text{rank}(A')\le k} ||A - A'|| = \sigma_{k+1}(A)$$

其中 $\sigma_{r+1}(A) = 0$。实际上，这个估计 $A_k \sim A$ 对上述关系在任何酉不变矩阵范数下均成立（不做证明），即我们也可以把上述的矩阵范数换成 $||\cdot||_F$ 或者 $||\cdot||_*$。A 与 A_k 的关系如下（奇异值顺序递增）显然每一项得到的矩阵的秩均为 1：

$$\begin{aligned} A &= \sum_{i=1}^k \sigma_i \begin{bmatrix} * \\ \vdots \\ * \end{bmatrix}_i \begin{bmatrix} * & \cdots & * \end{bmatrix}_i + \sum_{i=k+1}^r \sigma_i \begin{bmatrix} * \\ \vdots \\ * \end{bmatrix}_i \begin{bmatrix} * & \cdots & * \end{bmatrix}_i \\ &= A_k + \sum_{i=k+1}^r \sigma_i \begin{bmatrix} * \\ \vdots \\ * \end{bmatrix}_i \begin{bmatrix} * & \cdots & * \end{bmatrix}_i \end{aligned}$$

实际上主成分分析 PCA 也是利用了这个原理。比如我们将 n 个 m 维数据放在 $m \times n$ 矩阵 A 中，如果想要降至 k 维，我们提取 A_k，将 U_k 的各列作为 k 个新的方向，将 $\Sigma_k V_k^H$ 的各列（下面公式的红色部分）作为新的数据点（在 k 个新方向 $u_{\cdot i}$ 上的坐标），即可将维度从 m 降至 k

$$A_k = \begin{bmatrix} u_{\cdot 1} & \cdots & u_{\cdot k} \end{bmatrix} \begin{bmatrix} \sigma_1 v_{11} & \cdots & \sigma_1 v_{1n} \\ \vdots & \vdots & \vdots \\ \sigma_k v_{k1} & \cdots & \sigma_k v_{kn} \end{bmatrix}$$

Proof Theorem 2.3.1. **(1)** 对 $\forall : ||\boldsymbol{x}|| = 1$，考虑 A 的 full SVD s.t. $A = U\Sigma V^T$，由向量 2-范数的酉不变性质，有

$$\begin{aligned} ||A\boldsymbol{x}|| &= ||U\Sigma V^T \boldsymbol{x}|| = ||\Sigma(V^T \boldsymbol{x})|| = ||\Sigma \boldsymbol{y}|| \\ &= \sqrt{\textstyle\sum_{i=1}^r \sigma_i^2 y_i^2} \le \sqrt{\textstyle\sum_{i=1}^r \sigma_1^2 y_i^2} \le \sigma_1 \cdot \sqrt{\textstyle\sum_{i=1}^n y_i^2} = \sigma_1 ||\boldsymbol{y}|| = \sigma_1 \end{aligned}$$

取 $\boldsymbol{x} = v_{\cdot 1}$ 同理可证（2），此处略过。

(3) 考虑 A 的 full SVD s.t. $A = U\Sigma V^T$

$$\begin{aligned} ||A||_F^2 &= \text{trace}(A^H A) = \text{trace}(V\Sigma^T \Sigma V^T) \\ &= \text{trace}(\Sigma V^T V \Sigma^T) = \text{trace}(\Sigma\Sigma^T) = \textstyle\sum_{i=1}^r \sigma_i^2 \end{aligned}$$

Proof Theorem 2.3.2 任取 $A' \in \mathbb{C}^{m\times n}$ s.t. $\text{rank}(A') \le k \le r$，由 dimension theorem，$\text{nullity}(A') = n - \text{rank}(A') \ge n - k$，即存在 ONB（规范正交基底）$W \in \mathbb{C}^{n\times(n-k)}$ s.t. $A'W = 0$ 且 $W^H W = I$。注意到此时 W 和 V_{k+1} 的列恰好构成 $\begin{bmatrix} W & V_{k+1} \end{bmatrix} \in \mathbb{C}^{n\times(n+1)}$，所以这些列必然线性相关。即：$\exists \boldsymbol{x}_1$，$\boldsymbol{x}_2 \neq \boldsymbol{0}$ s.t. $W\boldsymbol{x}_1 - V_{k+1}\boldsymbol{x}_2 = 0$，我们可以对 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 的长度进行放缩，使得有 $||W\boldsymbol{x}_1|| = ||V_{k+1}\boldsymbol{x}_2|| = ||\boldsymbol{x}_1|| = ||\boldsymbol{x}_2|| = 1$（矢量 2-范数的酉不变性）。取这么一个 $\boldsymbol{x}_1$，我们有：

$$\begin{aligned} ||A - A'|| &\ge ||(A - A')W|| = ||AW|| = ||U\Sigma(V^H W)|| \\ &\ge ||U\Sigma(V^H W)\boldsymbol{x}_1|| = ||\Sigma V^H V_{k+1}\boldsymbol{x}_2|| \\ &= ||\Sigma_{k+1}\boldsymbol{y}_2|| \ge \sigma_{k+1}(A) \end{aligned}$$

第一个不等号是因为，由矢量乘法的酉不变性，对于任意的列规范正交矩阵 W 和一个适配乘法的矩阵 B，$\{||BW\boldsymbol{x}|| : ||\boldsymbol{x}|| = 1\} \subseteq \{||B\boldsymbol{x}|| : ||\boldsymbol{x}|| = 1\}$。因此集合上的最大值应当符合关系 $||BW|| \le ||B||$。最后 $||A - A'|| \ge \sigma_{k+1}(A)$ 在 $A' = A_k$ 时取等号，证毕。

2.4 Courant-Fisher Minimax Theorem

Theorem 2.4.1. Courant-Fisher Minimax Theorem（CF）
记矩阵 $A \in \mathbb{C}^{m\times n}$，其秩 $\text{rank}(A) = r$，$S_n \subseteq \mathbb{C}^n$ 为一子（线性）空间，则

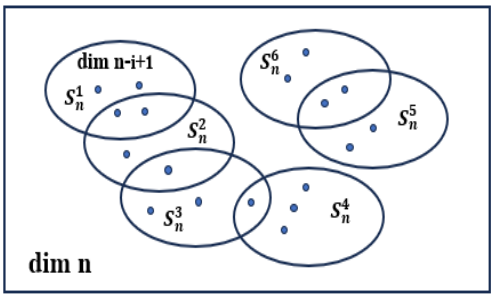
$$\begin{aligned}\sigma_i(A) &= \min_{\dim(S_n)=n+1-i} \max_{\boldsymbol{x}\in Sn\ ,\ ||\boldsymbol{x}||=1} ||A\boldsymbol{x}|| \\ &= \max_{\dim(S_n)=i} \min_{\boldsymbol{x}\in Sn\ ,\ ||\boldsymbol{x}||=1} ||A\boldsymbol{x}||\end{aligned}$$

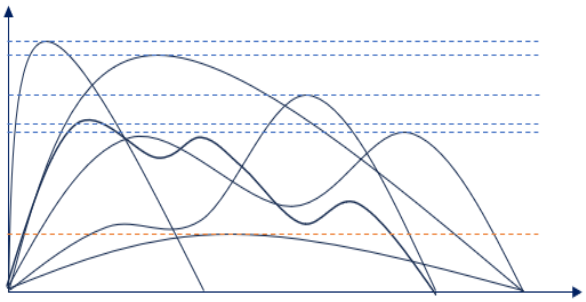
可以理解为所有 $n+1-i$ 维子空间中，矩阵能实现最大幅度变换的下限；或者，所有 i 维子空间中，矩阵能实现最小幅度变换的上限。注意，对于集合 $\{||A\boldsymbol{x}|| : \boldsymbol{x} \in Sn\ ,\ ||\boldsymbol{x}|| = 1\ ,\ \dim(S_n) = i\}$（第二个最值的部分），我们总可以有如下的替换

$$\begin{aligned}&\{||A\boldsymbol{x}|| : \boldsymbol{x} \in Sn\ ,\ ||\boldsymbol{x}|| = 1\ ,\ \dim(S_n) = i\} \\ =\ &\{||AQ\boldsymbol{y}|| : \boldsymbol{y} \in \mathbb{C}^i\ ,\ ||\boldsymbol{y}|| = 1\ ,\ Q \in \mathbb{C}^{n\times i}\ ,\ Q^HQ = I_i\}\end{aligned}$$

类似地，对于降序排列的对称矩阵 A的特征值

$$\lambda_i(A) = \min_{\dim(S_n)=n+1-i} \max_{\boldsymbol{x}\in Sn} \frac{\boldsymbol{x}^TA\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \max_{\dim(S_n)=i} \min_{\boldsymbol{x}\in Sn} \frac{\boldsymbol{x}^TA\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}$$





Proof Theorem 2.4.1 我们先证明后一个等号。考虑 A 的 full SVD $A = U\Sigma V^H$, 令 $V_i^* = [v_{\cdot i}\cdots v_{\cdot n}]$ 观察到 $\dim(S_n)+\dim(V_i^*) = n-i+1+i = n+1 \geq n$, 所以 $\exists \boldsymbol{w} \in S_n \cap \text{span}(V_i^*)$ s.t. $\boldsymbol{w} = V_i^*\boldsymbol{y}$ 且 $||\boldsymbol{w}|| = ||\boldsymbol{y}|| = 1$。于是对任意的 $\dim(S_n) = i$，总存在 $||\boldsymbol{w}|| = ||\boldsymbol{y}|| = 1$ 使得

$$||Aw|| = ||U\Sigma V^Hw|| = ||\Sigma V^HV_i^*\boldsymbol{y}|| = ||\text{diag}(\sigma_i\cdots\sigma_n)\boldsymbol{y}|| \leq \sigma_i$$

即，对任意的 $\dim(S_n) = i$，$\sigma_i \geq ||Aw|| \geq \min_{\boldsymbol{x}\in Sn\ ,\ ||\boldsymbol{x}||=1} ||A\boldsymbol{x}||$。当取 $S_n = V_i = [v_{\cdot 1}\cdots v_{\cdot i}]$ 时，等号成立。前一个等号及对称矩阵特征值相关的证明与这个完全类似，前者取 $V_i = [v_{\cdot 1}\cdots v_{\cdot i}]$；后者用规范正交对角化的 Q 取代前面的 V 即可。

Theorem 2.4.2. Weyl's Theorem

1) 任何矩阵 A 即期奇异值: $\left|\ \sigma_i(A+E)-\sigma_i(A)\ \right| \ \leq\ ||E||_2=\sigma_1(E)$

2) 当 $i=1$: $||A||_2-||E||_2\ \leq\ ||A+E||_2\ \leq\ ||A||_2+||E||_2$

3) 实对称矩阵的特征值: $\left|\ \lambda_i(A+E)-\lambda_i(A)\ \right| \ \leq\ ||E||_2$

这基本说明奇异值和对称矩阵的特征值对于扰动不明感（*well-conditioned*）。

Theorem 2.4.3. 奇异值的额外性质

1) $\sigma_i(A^H)=\sigma_i(A)\ \ \text{if}\ \ 1\leq i\leq \text{rank}(A)=\text{rank}(A^H)$

2) $\sigma_{\min(m,n)}(A)\sigma_i(B)\leq\sigma_i(AB)\leq\sigma_1(A)\sigma_i(B)\ \ \text{if}\ \ A\in\mathbb{C}^{m\times n}$

3) $\sigma_i(A)\sigma_{\min(n,k)}(B)\leq\sigma_i(AB)\leq\sigma_i(A)\sigma_1(B)\ \ \text{if}\ \ B\in\mathbb{C}^{n\times k}$

4) $\sigma_i\left(\begin{bmatrix}A_1\\A_2\end{bmatrix}\right)\geq\max\left(\sigma_i(A_1)\ ,\ \sigma_i(A_2)\right)$

5) $\sigma_i\left(\begin{bmatrix}A_1 & A_2\end{bmatrix}\right)\geq\max\left(\sigma_i(A_1)\ ,\ \sigma_i(A_2)\right)$

后两个性质证明见 Lecture Notes p.23

Proof Theorem 2.4.2 (1) 我们只证明 $\sigma_i(A+E)\leq\sigma_i(A)+||E||_2$，另一侧的证明（包括特征值的情形）均与之类似。记 $S_n^i\subseteq\mathbb{C}^n$ ，$\dim(S_n^i)=i$

$$
\begin{aligned}
\sigma_i(A+E)&=\max_{S_n^i}\left\{\min_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}||\left(A+E\right)\boldsymbol{x}||\right\}\\
&\leq\max_{S_n^i}\left\{\min_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}\left(||A\boldsymbol{x}||+||E\boldsymbol{x}||\right)\right\}\\
&\leq\max_{S_n^i}\left\{\min_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}||A\boldsymbol{x}||+\max_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}||E\boldsymbol{x}||\right\}\\
&\leq\max_{S_n^i}\min_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}||A\boldsymbol{x}||+\max_{S_n^i}\max_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}||E\boldsymbol{x}||\\
&\leq\max_{S_n^i}\min_{\boldsymbol{x}\in S_n^i\ ,\ ||\boldsymbol{x}||=1}||A\boldsymbol{x}||+\max_{||\boldsymbol{x}||=1}||E\boldsymbol{x}||=\sigma_i(A)+||E||_2
\end{aligned}
$$

Proof Theorem 2.4.3 (2) (1) 显然，我们只证明 (2)/(3)。记 $A \in \mathbb{C}^{m\times n}$，$B \in \mathbb{C}^{n\times k}$，$S_i \subseteq \mathbb{C}^k$，$\dim(S_i) = i$，我们有关系：

$$\min_{\boldsymbol{x}\in S_i} \frac{||AB\boldsymbol{x}||}{||B\boldsymbol{x}||}\frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \geq \min_{\boldsymbol{x}\in S_i} \frac{||AB\boldsymbol{x}||}{||B\boldsymbol{x}||} \cdot \min_{\boldsymbol{x}\in S_i} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \geq \sigma_{\min(m,n)}(A) \min_{\boldsymbol{x}\in S_i} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||}$$

$$\max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||AB\boldsymbol{x}||}{||B\boldsymbol{x}||}\frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \leq \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||AB\boldsymbol{x}||}{||B\boldsymbol{x}||} \cdot \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \leq \sigma_1(A) \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||}$$

于是，对 $AB \in \mathbb{C}^{m\times k}$，有

$$\begin{aligned}
\sigma_i(AB) &= \max_{S_i} \min_{\boldsymbol{x}\in S_i} \frac{||AB\boldsymbol{x}||}{||\boldsymbol{x}||} = \max_{S_i} \min_{\boldsymbol{x}\in S_i} \frac{||AB\boldsymbol{x}||}{||B\boldsymbol{x}||}\frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \\
&\geq \max_{S_i} \left\{ \sigma_{\min(m,n)}(A) \min_{\boldsymbol{x}\in S_i} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \right\} \\
&= \sigma_{\min(m,n)}(A) \max_{S_i} \min_{\boldsymbol{x}\in S_i} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} = \sigma_{\min(m,n)}(A)\sigma_i(B)
\end{aligned}$$

$$\begin{aligned}
\sigma_i(AB) &= \min_{S_{k+1-i}} \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||AB\boldsymbol{x}||}{||\boldsymbol{x}||} = \min_{S_{k+1-i}} \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||AB\boldsymbol{x}||}{||B\boldsymbol{x}||}\frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \\
&\leq \min_{S_{k+1-i}} \left\{ \sigma_1(A) \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} \right\} \\
&= \sigma_1(A) \min_{S_{k+1-i}} \max_{\boldsymbol{x}\in S_{k+1-i}} \frac{||B\boldsymbol{x}||}{||\boldsymbol{x}||} = \sigma_1(A)\sigma_i(B)
\end{aligned}$$

故（2）证毕，有

$$\sigma_{\min(m,n)}(A)\sigma_i(B) \leq \sigma_i(AB) \leq \sigma_1(A)\sigma_i(B)$$

要证（3），对 AB 取共轭转置即可

$$\sigma_{\min(n,k)}(B^H)\sigma_i(A^H) \leq \sigma_i(B^H A^H) \leq \sigma_1(B^H)\sigma_i(A^H)$$

由（1）$\sigma_i(B^H A^H) = \sigma_i(AB)$，$\sigma_i(A^H) = \sigma_i(A)$ 与 $\sigma_i(B^H) = \sigma_i(B)$ 易得

$$\sigma_{\min(n,k)}(B)\sigma_i(A) \leq \sigma_i(AB) \leq \sigma_1(B)\sigma_i(A)$$

3 LU Factorisation

本章中我们将讨论如何利用矩阵分解求线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$。其思路是对系数矩阵 A 进行分解 $A=BC$，其中 B，C 是两个结构相对简单的矩阵。接着通过分别求解 $B\boldsymbol{y}=\boldsymbol{b}$ 与 $C\boldsymbol{x}=\boldsymbol{y}$ 得到方程组的解 $\boldsymbol{x}$。在本章的后续内容中，我们将主要讨论矩阵的 (P)LU 分解。

3.1 Introduction to LU Factorisation

在之前所有的解线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$ 的实践中，我们的通用步骤是使用行初等变换，i.e., 高斯消元法，从而将线性方程组 $A\boldsymbol{x}=\boldsymbol{b}$ 转化为等价的 $U\boldsymbol{x}=\tilde{\boldsymbol{b}}$，使得新的系数矩阵 U 为一个上三角矩阵（不一定 square）。我们知道，所有的行初等变换实际上都可以通过左乘一个行初等变换矩阵实现：

Definition 3.1.1. 行初等变换矩阵 作用在矩阵 $A\in\mathbb{F}^{m\times n}$ 上的所有的三类行初等变换，*i.e.*, 倍加 *replacement*，互换 *interchange*，伸缩 *scaling* 都可以通过左乘一个行初等变换矩阵 $E\in\mathbb{F}^{m\times m}$ 得到。定义如下的记号

变换	作用	记号	逆矩阵
倍加	将第 i 行 $\times k$ 加到第 j 行	$E_{ij}(k)$	$E_{ij}(-k)$
互换	将第 i 行与第 j 行互换位置	E_{ij}	$E_{ji}=E_{ij}$
伸缩	将第 i 行 $\times k$	$E_i(k)$	$E_i(k^{-1})$

行初等变换矩阵可以通过对单位矩阵 I_m 执行其所对应的操作得到。更进一步，行初等矩阵的乘法结果 $E_n\cdots E_1$ 可以通过对 I_m 依次执行 $E_1\sim E_n$ 的操作得到。用同样的方法可以得到列初等变换矩阵（列初等变换等价于右乘列初等矩阵）。

于是高斯消元法等价于同时在方程的两侧同时左乘若干（有限）个行初等变换矩阵 $E_1\sim E_n$

$$E_n\cdots E_2E_1A\boldsymbol{x}=U\boldsymbol{x}=E_n\cdots E_2E_1\boldsymbol{b}\quad\Rightarrow\quad E_n\cdots E_2E_1A=U$$

因此我们有如下关于 A 的分解

$$A=E_n^{-1}\cdots E_2^{-1}E_1^{-1}U=LU\quad\text{s.t.}\quad L=E_n^{-1}\cdots E_2^{-1}E_1^{-1}$$

如果我们对高斯消元的流程作如下限制：不涉及行互换操作，且倍加 $E_{ij}(k)$ 时总有 $i<j$，如果在此流程下仍能得到预期的上三角矩阵 U（注意这不一定总是可行的），则我们有上述的 $E_1\sim E_n$（及其逆矩阵）为一系列下三角矩阵，从而 L 也是一个下三角矩阵。我们称 $A=LU$ 为矩阵 A 的 LU 分解。

Definition 3.1.2. LU Factorisation
对于矩阵 $A \in \mathbb{F}^{m\times n}$，若存在 $A = LU$ 使得 $L \in \mathbb{F}^{m\times m}$ 为下三角矩阵；$U \in \mathbb{F}^{m\times n}$ 为上三角矩阵，则称该分解为 A 的 LU 分解。LU 分解存在的一个充分条件是可以通过不涉及行互换的高斯消元法得到上三角矩阵 U。

Example 3.1.1. LU 分解不存在的情形 尝试使用高斯消元法对下列矩阵进行 LU 分解

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

首先进行行初等变换

$$A = \begin{bmatrix} 1 & 2 & -3 & 1 \\ 2 & 4 & 0 & 7 \\ -1 & 3 & 2 & 0 \end{bmatrix} \xrightarrow{-2r_1 + r_2} \begin{bmatrix} 1 & 2 & -3 & 1 \\ 0 & 0 & 6 & 5 \\ -1 & 3 & 2 & 0 \end{bmatrix} \xrightarrow{r_1 + r_3}$$

$$\begin{bmatrix} 1 & 2 & -3 & 1 \\ 0 & 0 & 6 & 5 \\ 0 & 5 & -1 & 1 \end{bmatrix} \xrightarrow{r_2 \leftrightarrow r_3} \begin{bmatrix} 1 & 2 & -3 & 1 \\ 0 & 5 & -1 & 1 \\ 0 & 0 & 6 & 5 \end{bmatrix} = U$$

即我们有 $E_3E_2E_1\ A = E_{23}\cdot E_{13}(1)\cdot E_{12}(-2)\ A = U$，即

$$\tilde{L} = (E_3E_2E_1)^{-1} = E_{12}(-2)^{-1}\cdot E_{13}(1)^{-1}\cdot E_{23}^{-1} = E_{12}(2)\cdot E_{13}(-1)\cdot E_{23}$$

将对应的行初等变换矩阵从右往左作用于单位矩阵 I 得到

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow{r_2 \leftrightarrow r_3} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \xrightarrow{-r_1 + r_3}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \xrightarrow{2r_1 + r_2} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} = \tilde{L}$$

可以发现，由于在高斯消元的过程中使用了行互换，导致初等变换矩阵乘积的逆并不是一个下三角矩阵，因此这个矩阵没有 LU 分解。但经过观察，我们总是可以找到一个排列矩阵 P 使得 $P\tilde{L} = L$ 为下三角矩阵，从而得到 $PA = P\tilde{L}U = LU$ 为 A 的 PLU 分解。

Definition 3.1.3. 排列矩阵（permutation matrix） 对于一个给定的排列 $\tau(i) : \{1 \cdots n\} \mapsto \{1 \cdots n\}$，作用在矩阵 $A \in \mathbb{F}^{m \times n}$ 上的排列矩阵为

$$P = \begin{bmatrix} \boldsymbol{e}_{\tau(1)}^T \\ \vdots \\ \boldsymbol{e}_{\tau(n)}^T \end{bmatrix}$$

其中 $\boldsymbol{e}_i$ 为第 i 个单位向量。左乘 P 相当于对行做相应的排列；右乘 P^T 相当于对列做相应的排列。由简单的抽象代数知识，所有 *permutation* 都可以分解成若干个二元 *transpose* 的复合，故而 P 实际上是一系列（行/列）互换初等矩阵 E_{ij} 的复合。显然，所有的排列矩阵 P 都是 orthogonal matrix $P^T P = P P^T = I$。

Definition 3.1.4. PLU Factorisation
对于矩阵 $A \in \mathbb{F}^{m \times n}$，使用高斯消元法总可以将其转化为 $PA = LU$ 的形式，使得 $P \in \mathbb{F}^{m \times m}$ orthogonal 为一个规范正交排列矩阵；$L \in \mathbb{F}^{m \times m}$ 为下三角矩阵；$U \in \mathbb{F}^{m \times n}$ 为上三角矩阵，称该分解为 A 的 PLU 分解。此时 $A\boldsymbol{x} = \boldsymbol{b}$ 等价于求 $LU\boldsymbol{x} = P\boldsymbol{b}$。有时称这一分解为 A 的 *Pivoted LU Factorisation*。

下面给出利用高斯消元法手动求解矩阵 A 的 PLU 分解的具体流程

PLU Factorisation Using Gaussian Elimination

STEP 1 对 A 照常进行 Gaussian Elimination，得到 U 并记录下每一步对应的行初等变换矩阵 E_i s.t.

$$E_n \cdots E_1 A = U$$

STEP 2 对 $E_n \cdots E_1$ 取逆，并将逆变换 $E_1^{-1} \cdots E_n^{-1}$ 对应的行操作执行于单位矩阵 I 得到 $\tilde{L} = PL$ s.t.

$$E_1^{-1} \cdots E_n^{-1} I = \tilde{L}$$

STEP 3 将 $\tilde{L}$ 整理成下三角得到 L，对比二者的行得到 P 对应的排列 τ 使得 $\tau(\text{row index in } \tilde{L}) = \text{row index in } L$ 即可，根据 τ 得到对应的排列矩阵 P_τ 使得 $P_\tau \tilde{L} = L$，最后有

$$P_\tau A = LU$$

3.2 Algorithm for LU Factorizsation

本节介绍如何使用规范化的数值方法对矩阵 A 进行 (R)LU 分解。首先我们介绍一种最简单的 LU 分解作为引入，暂时先不妨假设矩阵 A 存在 LU 分解且为 n 阶方阵。 为了统一流程，做如下的记号规定：

记 $R_1 := A \in \mathbb{F}^{n\times n}$，数域可为实数或复数域；对 R_1，即 A，以及随后递推产生的所有 $\{R_i\}_{i=1}^n$（n 等于矩阵 A 的阶），用 L_i 和 U_i^T 分别表示他们的第 i 列/行；用 r_i 表示 R_i 的第 (i,i) 号元素（由推导也可以看出是 R_i 非零子阵的左上顶点元素）。我们用如下递推关系产生出余下的 $\{R_i\}_{i=1}^n$

$$i = 1\cdots(n-1) \quad : \quad R_{i+1} = R_i - r_i^{-1}L_iU_i^T$$

Lemma 3.2.1. *在如上记号规定下，若 $A \in \mathbb{F}^{n\times n}$ 存在 LU 分解，则*

$$R_{n+1} := R_n - r_n^{-1}L_nU_n^T = 0$$

上述引理是一个显然的结果，实际上，由于 $r_1^{-1}L_1U_1^T$ 产生的矩阵的第一行一列与 R_1，也就是 A 完全相同。这导致 R_2，也即 R_1 与 $r_1^{-1}L_1U_1^T$ 的差矩阵的第一行一列为全 0 元素。实际上由归纳法不难看出，若用 $*$ 表示矩阵位置上可能不为 0 的元素，R_i 与 $r_i^{-1}L_iU_i^T$ 分别有如下结构

$$R_i = \begin{bmatrix} O & O & O & O \\ O & r_i & \cdots & r_{in} \\ O & \vdots & * & * \\ O & r_{ni} & * & * \end{bmatrix}$$

$$r_i^{-1}L_iU_i^T = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ r_i/r_i \\ \vdots \\ r_{ni}/r_i \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 & r_i & \cdots & r_{in} \end{bmatrix} = \begin{bmatrix} O & O & O & O \\ O & r_i & \cdots & r_{in} \\ O & \vdots & *' & *' \\ O & r_{ni} & *' & *' \end{bmatrix}$$

每一次迭代作差，都会使一行一列消失变为 0，直至到 R_n 时有 $R_n = r_n^{-1}L_nU_n^T$。之所以要求 A 存在 LU 分解，是因为需要确保总有 $r_i > 0$。s 上述引理也可以推广到非方阵的情形：

Lemma 3.2.2. 在如上记号规定下，若 $A \in \mathbb{F}^{m\times n}$ 存在 LU 分解，则

$$d = \min(m,n) \quad \text{s.t.} \quad R_{d+1} := R_d - r_d^{-1} L_d U_d^T = 0$$

在上述引理的支持下，有如下的 LU 分解定理及算法

Theorem 3.2.1. Algorithm for LU Factorisation
在如上记号规定下，若 $A \in \mathbb{F}^{m\times n}$ 存在 LU 分解，$d = \min(m,n)$，则有

$$A = LU = \begin{bmatrix} L_1/r_1 & \cdots & L_d/r_d \end{bmatrix} \begin{bmatrix} U_1^T \\ \vdots \\ U_d^T \end{bmatrix}$$

给出如下伪代码

Algorithm 1: LU Factorisation

Initialize: $d \longleftarrow \min(m,n)$
Initialize: $R_1 \longleftarrow A$; $L \longleftarrow [\,]$; $U \longleftarrow [\,]$

```
for i = 1 : d do
    r_i , L_i , U_i ⟵ R_i(i,i) , R_i(:,i) , R_i(i,:)
    if r_i ≠ 0 then
        R_{i+1} ⟵ R_i − r_i^{-1} L_i U_i^T
    Append r_i^{-1} L_i to L as columns ; append U_i^T to U as rows
```

Proof Theorem 3.2.1 由递推关系，我们有

$$\begin{aligned} R_2 &= R_1 - r_1^{-1} L_1 U_1^T \\ R_3 &= R_2 - r_2^{-1} L_2 U_2^T \\ &\vdots \\ R_{d+1} &= R_d - r_d^{-1} L_d U_d^T \end{aligned}$$

两侧相加，又有 $R_1 = A$，$R_{d+1} = 0$，于是

$$A = r_1^{-1} L_1 U_1^T + \cdots + r_d^{-1} L_d U_d^T = \begin{bmatrix} L_1/r_1 & \cdots & L_d/r_d \end{bmatrix} \begin{bmatrix} U_1^T \\ \vdots \\ U_d^T \end{bmatrix} = LU$$

显然 L, U 分别为下三角和上三角矩阵。

3.3 Further Improvements for LU Factorisations

对于上一节的 Algorithm 1, 如果为 n 阶方阵, 第 i 个循环要实际执行 $n-i+1$ 步除法, $(n-i)^2$ 步（矩阵）乘法与 $(n-i)^2$ 步（矩阵）减法, 总计 n 步, 时间复杂度约为 $\frac{2}{3}n^3 \sim O(n^3)$。不过显然, 上述讨论的算法是有缺陷的, 比如并不是所有的矩阵都存在 LU 分解。我们可以使用一种名为 Pivoting 的技巧在上述算法的基础上稍加改进, 从而实现对任何可逆矩阵都能执行的 PLU 分解。

假设 $A \in \mathbb{F}^{n\times n}$ 为可逆方阵, 仍然记 L_i, U_i^T 和 r_i 分别为 $\{R_i\}_{i=1}^n$ 的第 i 列/行以及 第 (i,i) 号元素。不过这次对 R_i 的递推关系做如下改进:

$$i = 1\cdots(n-1) \quad : \quad R_{i+1} = P_{i+1}\left(R_i - r_i^{-1}L_iU_i^T\right) \quad \text{where} \quad R_1 = P_1A$$

其中 P_i 为一个行互换初等矩阵 $P_i = E_{ij}$ for $j \geq i$, 它将原本余项 $R_i' = R_{i-1} - r_{i-1}^{-1}L_{i-1}U_{i-1}^T$ 中的 leading pivot r_i' 换为其所在列的（modulus）最大元素 r_j' s.t. $j \geq i$。这一步骤称为 Pivoting。在此基础上, 我们有

$$\begin{aligned}
R_1 &= P_1A \\
R_2 &= P_2(R_1 - r_1^{-1}L_1U_1^T) \\
R_3 &= P_3(R_2 - r_2^{-1}L_2U_2^T) \\
&\vdots \\
R_n &= P_n(R_{n-1} - r_{n-1}^{-1}L_{n-1}U_{n-1}^T) = r_n^{-1}L_nU_n^T
\end{aligned}$$

于是, 从最后一个式子开始回代, 我们有

$$P_n\cdots P_2P_1\ A = \sum_{i=1}^{n-1} P_nP_{n-1}\cdots P_{i+1}r_i^{-1}L_iU_i^T + r_n^{-1}L_nU_n^T$$

$$PA = \begin{bmatrix} L_1'/r_1 & \cdots & L_n'/r_n \end{bmatrix} \begin{bmatrix} U_1^T \\ \vdots \\ U_d^T \end{bmatrix} = LU$$

其中有 $P = P_n\cdots P_2P_1$, 以及 $L_i' = P_nP_{n-1}\cdots P_{i+1}L_i$, $L_n' = L_n$。由于对列向量 L_i, $P_k = E_{kj}$ for $j \geq k$ 只影响 j,k 两行, 而 $j \geq k \geq i+1 > i$。即 $P_nP_{n-1}\cdots P_{i+1}$ 作用于 L_i 不会改变其形状（从第 i 个元素开始不为 0）。因此 L 仍然是一个下三角矩阵。此外显然 $P = P_n\cdots P_2P_1$ 是一个排列矩阵。

Theorem 3.3.1. Algorithm for Pivoted LU Factorisation
在如上记号规定下，若 $A \in \mathbb{F}^{n\times n}$ 为可逆方阵，总存在 PLU 分解

$$PA = \begin{bmatrix} L'_1/r_1 & \cdots & L'_n/r_n \end{bmatrix} \begin{bmatrix} U_1^T \\ \vdots \\ U_d^T \end{bmatrix} = LU$$

其中有 $P = P_n \cdots P_2 P_1$，以及 $L'_i = P_n P_{n-1} \cdots P_{i+1} L_i$ ，$L'_n = L_n$。P_i 为一个行互换初等矩阵 $P_i = E_{ij}$ for $j \geq i$，将原本余项 $R'_i = R_{i-1} - r_{i-1}^{-1} L_{i-1} U_{i-1}^T$ 中的 *leading pivot* r'_i 换为其所在列的（*modulus*）最大元素 r'_j s.t. $j \geq i$。算法伪代码如下

Algorithm 2: Pivoted LU Factorisation (PLU)

Initialize: $R \longleftarrow P_1 A$; $P \longleftarrow I_n$
Initialize: $L \longleftarrow [\,]$; $U \longleftarrow [\,]$

1 **for** $i = 1 : n$ **do**
2 r_i , L_i , U_i $\longleftarrow R(i,i)$, $R(:,i)$, $R(i,:)$
3 Append $r_i^{-1} L_i$ to L as columns ; append U_i^T to U as rows
4 **if** $i < n$ **then**
5 $R \longleftarrow R - r_i^{-1} L_i U_i^T$
6 $R \longleftarrow P_{i+1} R$
7 $P \longleftarrow P_{i+1} P$
8 **for** *col* **in** L **do**
9 $col \longleftarrow P_{i+1} col$
10 $P \longleftarrow P P_1$

注意上述伪代码中，P_1 必须保留至最后，因为在第十行之前，循环结束后伪代码给出的 $P = P_n \cdots P_2$。在第六行之前，也最好检查一下目标列是不是全 0 列。该算法的复杂度（不考虑遍历寻找 P_i）仍然为 $\frac{2}{3}n^3 \sim O(n^3)$。使用 Pivoting 后算法稳定性有所提升，但仍可能出现不稳定的情形。

当矩阵 $A \in \mathbb{R}^{n\times n}$ 为正定实对称方阵时，我们可以进一步简化算法。一方面，此时 A 必满秩且能 LU 分解，无需 Pivoting；另一方面，对每一步更新，我们都有余项 R_i 的第 i 行列 L_i，U_i 相同，记 $V_i := L_i/\sqrt{r_i} = U_i/\sqrt{r_i}$，所以对余项的更新可以进一步简化为

$$i = 1 \cdots (n-1) \quad : \quad R_{i+1} = R_i - V_i V_i^T$$

于是给出如下的改进算法

Theorem 3.3.2. Cholesky Factorisation
若 $A \in \mathbb{R}^{n\times n}$ 为正定实对称方阵，记 $V_i := L_i/\sqrt{r_i} = U_i/\sqrt{r_i}$，则必有 LU 分解

$$A = \boldsymbol{V}\boldsymbol{V}^T = [V_1 \ \cdots \ V_n] \begin{bmatrix} V_1^T \\ \vdots \\ V_n^T \end{bmatrix}$$

可以证明，如 A 是正定的，则 R_i 的非零子阵 $R_i(i:, i:)$ 也是正定的 Lecture Notes p.35，这意味着总有 $r_i > 0$。给出如下伪代码

Algorithm 3: Cholesky Factorisation for $A \succ 0$

Initialize: $R_1 \longleftarrow A$; $\boldsymbol{V} \longleftarrow [\,]$

for $i = 1 : n$ **do**
- $V_i \longleftarrow R_i(:, i) / \sqrt{R_i(i, i)}$
- $R_{i+1} \longleftarrow R_i - V_i V_i^T$
- Append V_i to $\boldsymbol{V}$ as columns

此算法的复杂度约为 $\frac{1}{3}n^3 \sim O(n^3)$，是一般 LU 分解算法的一半左右。这是因为由于其对称性，我们每一次计算 $V_i V_i^T$ 和相应的矩阵减法都只需要计算相当于先前使用 $r_i^{(-1)} L_i U_i^T$ 时的一半。

4 QR Factorisation

4.1 Introduction to QR Factorisation

本章中我们将讨论如何对矩阵进行 QR 分解。和 SVM 一样，QR 分解对任意一个矩阵 $A \in \mathbb{F}^{m\times n}$ 都存在，其在处理超定方程（最小二乘法）以及解特征值问题等领域都有着广泛应用。

Definition 4.1.1. QR Factorisation
对任意一个矩阵 $A \in \mathbb{F}^{m\times n}$ s.t. $\operatorname{rank}(A) = r$ 若存在分解 $A = QR$ 使得有如下两种情况中的一种成立

1） $Q \in \mathbb{F}^{m\times n}$ 列规范正交 $Q^HQ = I_n$；$R \in \mathbb{F}^{n\times n}$ 为上三角方阵

2） $Q \in \mathbb{F}^{m\times m}$ unitary $Q^HQ = QQ^H = I_m$；$R \in \mathbb{F}^{m\times n}$ 为上三角矩阵

我们称前者为 A 的 *thin QR factorisation*，后者为 *full QR factorisation*。一般来说，和 full SVD 一样，full QR factorisation 有如下的形式

$$A = QR = \begin{bmatrix} Q_r & Q_{m-r}^{\perp} \end{bmatrix} \begin{bmatrix} R_r & O_1 \\ O_2 & O_3 \end{bmatrix}_{m\times n}$$

其中 $Q_{m-r}^{\perp}$ 取自 Q_r 的正交补空间。我们总是可以将 full QR 转为 thin QR，只需要取上述 full QR 中 R 的 n 阶顺序主子阵 $R_n := R(:n, :n)$ 和 Q 的前 n 列 $Q_n := Q(:, :n)$ 即可，于是我们有对应的 thin QR，即 $A = Q_nR_n$。

对矩阵进行 QR 分解的算法大致可以分为两类：三角正交化 Triangular Orthogonalisations 与 正交三角化 Orthogonal Triangularisations。在之前的学习中，我们已经接触了传统的 Gram-Schmidt 算法，其实际上属于 Triangular Orthogonalisations 的一种。

Definition 4.1.2. Triangular Orthogonalisations
此类算法的核心是对原矩阵 $A \in \mathbb{F}^{m\times n}$ 右乘一系列的可逆上三角方阵（一般是列初等变换矩阵的复合）$R_1 \sim R_d \in \mathbb{F}^{n\times n}$，使得其最终变成一个列规范正交矩阵 $Q \in \mathbb{F}^{m\times n}$

$$A\ R_1 \cdots R_d = Q \quad \Rightarrow \quad A = Q\ R_d^{-1} \cdots R_1^{-1} = QR$$

由于可逆三角矩阵的结构在乘法和取逆变换下保持稳定，我们可以得到 $A = QR$ 为一个 thin QR。

Definition 4.1.3. Orthogonal Triangularisations
此类算法的思路是对原矩阵 $A \in \mathbb{F}^{m\times n}$ 左乘一系列的酉矩阵 $Q_1 \sim Q_d \in \mathbb{F}^{m\times m}$，使得其最终变成一个列上三角矩阵 $R \in \mathbb{F}^{m\times n}$

$$Q_d \cdots Q_1 \; A = R \quad \Rightarrow \quad A = Q_1^H \cdots Q_d^H \; R = QR$$

酉矩阵的乘积仍然是酉矩阵，我们可以得到 $A = QR$ 为一个 full QR。

4.2 QR via Triangular Orthogonalisations

在本节中我们介绍两种 Triangular Orthogonalisation 算法，分别为 Classical Gram-Schmidt（CGS）与 Modified Gram-Schmidt（MGS）。其所使用的原理都是基于 Gram-Schmidt 正交化，只不过后者通过对算法流程的调整使得新算法的稳定性有了很大的提升。首先我们先来回顾一下 Classical Gram-Schmidt 的流程，其将 A 的列通过一系列的 列初等变换 转换为了一个列规范正交矩阵 Q。

Theorem 4.2.1. Classical Gram-Schmidt QR（CGS）
记 $A \in \mathbb{F}^{m\times n}$，且 $\text{rank}(A) = n$ 为列满秩矩阵 squar or tall。记 a_j 为 A 的第 j 列，采用如下的算法将 A 的各列转换为 orthonormal basis $\{q_j\}_{j=1}^n$

$$\begin{aligned} v_1 &= a_1 && ; \quad r_{11} = ||v_1|| \quad ; \quad q_1 = v_1/r_{11} \\ v_j &= a_j - \textstyle\sum_{i=1}^{j-1}\langle a_j, q_i\rangle q_i \quad ; \quad r_{jj} = ||v_j|| \quad ; \quad q_j = v_j/r_{jj} \end{aligned}$$

有 $r_{ij} = \langle a_j, q_i\rangle$，不难验证还有 $r_{jj} = \langle a_j, q_j\rangle$。该过程的每一步都相当于从当前处理的 a_i 中减去其在之前得到的规范正交基底 $q_1 \sim q_{i-1}$ 上的投影之后再进行一步规范化。这个流程也正是 *Classical Gram-Schmidt* 的流程，我们可以得到 $Q = [q_1 \cdots q_n]$ 与 $R = (r_{ij})$，从而 $A = QR$。

Algorithm 4: QR via CGS

```
for j = 1 : n do
    v_j ⟵ a_j
    for i = 1 : j − 1 do
        r_ij ⟵ ⟨a_j, q_i⟩
        v_j ⟵ v_j − r_ij q_i
    r_jj ⟵ ||v_j||
    q_j ⟵ v_j / r_jj
```

在上述 CGS 算法中，每个循环内需要进行 $(j-1)\times 4m + 4m = 4jm$ 次浮点运算（其中内积运算包含 m 次乘法和 m 次加法）。总计 n 个循环，故算法复杂度约为 $4m \times \frac{n(n+1)}{2} \sim O(2mn^2)$。然而，在下一章中，我们将会证明 CGS 是不稳定的，这意味着浮点误差将会随着迭代次数被逐渐放大。因此，实际应用中很少直接使用这一算法。

Theorem 4.2.2. Modified Gram-Schmidt QR（MGS）
记 $A \in \mathbb{F}^{m\times n}$，且 $\text{rank}(A) = n$ 为列满秩矩阵 squar or tall。记 a_j 为 A 的第 j 列，采用如下的算法将 A 的各列转换为 orthonormal basis $\{q_j\}_{j=1}^n$

Algorithm 5: QR via MGS

```
for j = 1 : n do
    v_j ⟵ a_j
for j = 1 : n do
    r_jj ⟵ ||v_j||
    q_j ⟵ v_j / r_jj
    for i = j + 1 : n do
        r_ij ⟵ ⟨v_i, q_j⟩
        v_i ⟵ v_i − r_ij q_j
```

在上述 MGS 算法只是对 CGS 的计算顺序进行了调整。在每一次一个新的基底 q_j 产生的时候，就马上将所有剩余列 v_i 中在 q_j 方向上的成分减去。由于之前每一次从 v_i 中删除的成分 $r_{ij'}q_{j'}$ 与新生成的基 q_j 总是正交的，用归纳法不难证明此处的 r_{ij} 和 CGS 中的 r_{ij} 一致。可以求出该算法的复杂度同样为 $O(2mn^2)$，但与 CGS 相比，MGS 大大提高了算法的稳定性。

不论是 CGS 还是 MGS，实际上都属于 triangular orthogonalisation。对 CGS，每一次（$j = 1\cdots n$）更新都相当于右乘上三角矩阵 R_j（见下页），R_j 相当于先进行 $j-1$ 个列 Replacement 变换后加上一个列 Scaling 变换的复合。只是对单位矩阵的第 i 列进行了替换。

对 MGS，每一次（$j = 1\cdots n$）更新都相当于右乘上三角矩阵 $\tilde{R}_j$（见下页），$\tilde{R}_j$ 相当于先进行一个列 Scaling 变换后再进行 $j-1$ 个列 Replacement 变换的复合，不过这次的 Replacement 是作用在后面的所有列上的。$\tilde{R}_j$ 的结构相当于对单位矩阵的第 i 行进行了替换。

$$R_j = \begin{bmatrix} 1 & \cdots & 0 & -r_{1j} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & -r_{j-1,j} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1/r_{jj} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad \widetilde{R}_j = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \frac{1}{r_{jj}} & -\frac{r_{j+1,j}}{r_{jj}} & \cdots & -\frac{r_{nj}}{r_{jj}} \\ 0 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

注意在上述算法中，一个自然的想法是在计算流程中不储存 r_{ij}，而在最后使用 $R = Q^T A$ 直接计算。注意这样做是有风险的，计算结果可能不稳定。

4.3 QR via Orthogonal Triangularisations

在本节中我们介绍两种 Orthogonal Triangularisation 算法，分别为 Householder 与 Given。后者在处理稀疏矩阵的 QR 分解时能节省很多的算力，从而对算法进行加速。二者都有着较高的稳定性，被广泛应用于几乎所有需要 QR 分解的场合。

Definition 4.3.1. Householder Reflectors
给定一个单位长度的实向量 $||v|| = 1$，实矩阵 $H_v = I - 2vv^T$ 为 v 对应的一个 *Householder Reflector*。不难证明其对称且 orthogonal。

Properties 4.3.1. 记 $H_v = I - 2vv^T$ 为一个 Householder Reflector，有
1) H_v 对称，orthogonal。
2) 对 $u//v$，H_v 将其翻转 $H_v u = -u$；对 $u \perp v$，H_v 使其保持原状 $H_v u = u$。
3) 若有 $||u|| = ||w||$，取 $v = \frac{w-u}{||w-u||}$ 总有 $H_v u = w$ 与 $H_v w = u$ (reflection)。

Proof Properties 4.3.1 (3) 前两个性质不难证明，省略，只证 $H_v u = w$。将 u 分解为 $u = u_{//} + u_\perp$ 两个分别与 v 平行，正交的成分，

$$H_v u = H_v(u_{//} + u_\perp) = -u_{//} + u_\perp = u - 2u_{//} = w - ||w-u||v - 2(u^T v)v$$

提取出余项部分，有

$$\begin{aligned} &-||w-u||v - 2(u^T v)v = -(||w-u||v + 2(u^T v))v \\ &= -\frac{v}{||w-u||}\{||w-u||^2 + 2u^T(w-u)\} = -\frac{v}{||w-u||}\{(w^T + u^T)(w-u)\} = 0 \end{aligned}$$

Theorem 4.3.1. 对任意的 $\boldsymbol{x} \in \mathbb{R}^m$，均存在 v_x^i 使得

$$v_x^i = \frac{\boldsymbol{x}_0^i \pm ||\boldsymbol{x}_0^i|| e_i}{||\boldsymbol{x}_0^i \pm ||\boldsymbol{x}_0^i|| e_i||} \quad \text{for} \quad \boldsymbol{x}_0^i = \begin{bmatrix} \mathbf{0}_{1:i-1} \\ \boldsymbol{x}_{i:m} \end{bmatrix} \quad \text{s.t.} \quad H_{v_x^i}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mp||\boldsymbol{x}_{i:m}|| \\ \mathbf{0}_{i+1:m} \end{bmatrix}$$

正负号一般取与 $\boldsymbol{x}_0^i$ 的第 i 个元素相同的符号，以避免 $\boldsymbol{x}_0^i \pm ||\boldsymbol{x}_0^i|| e_i = 0$ 的情况出现。上述的 e_i 表示第 i 个单位向量。

Proof Theorem 4.3.1

$$\begin{aligned} H_{v_x^i}\boldsymbol{x} &= H_{v_x^i}\left(\boldsymbol{x}_0^i + \begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mathbf{0}_{i:m} \end{bmatrix}\right) = H_{v_x^i}\boldsymbol{x}_0^i + H_{v_x^i}\begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mathbf{0}_{i:m} \end{bmatrix} \\ &= \mp||\boldsymbol{x}_0^i|| e_i + \left\{I - 2v_x^i(v_x^i)^T\right\}\begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mathbf{0}_{i:m} \end{bmatrix} \\ &= \mp||\boldsymbol{x}_0^i|| e_i + \begin{bmatrix} I_{i-1} & O \\ O & H_{v_{x_{i:m}}^1} \end{bmatrix}\begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mathbf{0}_{i:m} \end{bmatrix} \\ &= \mp||\boldsymbol{x}_0^i|| e_i + \begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mathbf{0}_{i:m} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_{1:i-1} \\ \mp||\boldsymbol{x}_{i:m}|| \\ \mathbf{0}_{i+1:m} \end{bmatrix} \end{aligned}$$

Theorem 4.3.2. Householder QR

记 $A \in \mathbb{R}^{m\times n}$ 为实矩阵，且 $d = \min(m, n)$。记 $X_1 := A$ 的第 1 列，x_j 为 $H_{j-1}\cdots H_1 X_1 = X_j$ 的第 j 列。有如下 orthogonal triangularisation

$$j = 1 \cdots d \ : \ v_j = v_{x_j}^j \quad ; \quad H_j = H_{v_j} \quad ; \quad X_j = H_{j-1}X_{j-1}$$

这最后将给我们 $R = H_d \cdots H_1 A$ 为上三角；$Q = (H_d \cdots H_1)^T = H_1^T \cdots H_d^T$ 为 orthogonal；$A = QR$ 为 A 的 full QR。

$$X_3 = \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \\ & & * & * \end{bmatrix} \xrightarrow{H_3} \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & & * \\ & & & * \end{bmatrix} = X_4 = H_3X_3 = H_3H_2H_1A$$

对于 n 阶方阵，Householder QR 的复杂度约为 $O\left(\frac{4}{3}n^3\right)$，是 GS 的两倍左右，但其大大提高了稳定性（Unconditionally stable，见下一章）；同时，它提供了完整的 QR 分解（full QR），因此也顺便提供了矩阵的秩。其最大的缺点是只能应用于实数矩阵。因为 Householder Refelctor 最重要的 property 4.3.1 (3) 对于复数内积会失效（复数内积 $w^H u \neq u^H w$）。

Householder QR 固然有效，但如果目标矩阵是一个稀疏矩阵，那么使用 Householder 将会浪费比较多的算力。因此我们引入下一种 Orthogonal Triangularisation 算法，称为 Givens QR。

Definition 4.3.2. Givens Rotations 引入如下一个 m 阶方阵，其除了在 $i<j$ ：(i,i) ，(i,j)，(j,i) ，(j,j) 四个位置上是单位矩阵，形如

$$G_{ij}(\theta)=\begin{bmatrix} I_{i-1} & & & & \\ & cos(\theta) & \cdots & sin(\theta) & \\ & \vdots & I_{j-i+1} & \vdots & \\ & -sin(\theta) & \cdots & cos(\theta) & \\ & & & & I_{m-j} \end{bmatrix}$$

在所有没有明确标出数值的地方都是 0（包括省略号部分），该矩阵称为 *Givens Rotation*。显然这是一系列正交矩阵，其作用于矩阵 $A\in\mathbb{F}^{m\times n}$ 的效果如下

$$G_{ij}(\theta)A=\begin{bmatrix} a_{1.}^T \\ \vdots \\ c_\theta a_{i.}^T+s_\theta a_{j.}^T \\ \vdots \\ c_\theta a_{j.}^T-s_\theta a_{i.}^T \\ \vdots \\ a_{m.}^T \end{bmatrix} \quad \text{s.t.} \quad c_\theta^2+s_\theta^2=1$$

注意 G_{ij} 不会改变前 $i-1$ 行的结构，所以我们可以利用 Givens Rotations 有针对性地对矩阵进行三角化，也即 *QR* 分解。

Theorem 4.3.3. Givens QR

记 $A \in \mathbb{R}^{m\times n}$ 为稀疏矩阵。观察第 j 列 a_j，当我们希望用 a_{jj} 将 a_{ij} 位置上的元素变为 0 时（一般 $i > j$），只需要找到合适的 θ 并应用 $G_{ji}(\theta)A$，于是此时，在 a_j 的第 i 行，我们有

$$c_\theta a_{ij} - s_\theta a_{jj} = 0 \quad \Rightarrow \quad tan(\theta) = \frac{a_{ij}}{a_{jj}} \quad \text{s.t.} \quad \theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$$

解得 $\theta_{ji} := \theta$ 即可。下面我们给出算法的整体流程伪代码

Algorithm 6: QR via Givens

Initialize: $R \longleftarrow A \ ; \ Q \longleftarrow I_m$

for $j = 1 : n$ **do**
  if $m \le j$ **then**
    Break
  for $i = j+1 : m$ **do**
    if $a_{ij} \neq 0$ **then**
      solve θ for $tan(\theta) = \frac{a_{ij}}{a_{jj}}$ s.t. $\theta \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
      $R \longleftarrow G_{ji}(\theta)\, R$
      $Q \longleftarrow G_{ji}(\theta)\, Q$
  $Q \longleftarrow Q^T$

注意在每开始处理新的一列 a_j 时，应用 G_{ji} 只会涉及 $i > j$ 的行，因此之前处理好的几列不会受到影响（在对应位置都已经是 0），具体细节如下

$$\begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & * & * \\ & & * & * \\ & & * & * \end{bmatrix} \xrightarrow{G_{34}} \begin{bmatrix} * & * & * & * \\ & * & * & * \\ & & *' & *' \\ & & 0 & *' \\ & & * & * \end{bmatrix}$$

Givens 也是一个稳定的算法，相较于 Householder，其优势在于可以局部调整减少算力，在面对稀疏矩阵时提高处理效率。

4.4 Least-squares Problems via QR

Definition 4.4.1. 超定系统与最小二乘问题
当线性方程组 $A\boldsymbol{x} = \boldsymbol{b}$ 中 $A \in \mathbb{F}^{m\times n}$ 为 tall $m >> n$ 时，我们称这个系统为超定的 *over determined*。这一般来说意味着所给的方程超出了求解所需的极限，因而此类方程往往不存在准确的解。此时，我们往往会将求解线性系统 $A\boldsymbol{x} = \boldsymbol{b}$ 转化为如下的优化问题

$$\min_{\boldsymbol{x}} ||A\boldsymbol{x} - \boldsymbol{b}||$$

目的是找到 $x^* = \text{argmin}_{\boldsymbol{x}}||A\boldsymbol{x} - \boldsymbol{b}||$，称此类问题为最小二乘问题 *least square problem*。

Definition 4.4.2. 最小二乘问题的解
在高等代数中，我们证明了最小二乘问题的解满足 $A^HA\boldsymbol{x}^* = A^H\boldsymbol{b}$（详见 *MTH107*, 此处不做展开）；当 $\text{rank}(A) = n$ 列满秩时（这一般对于超定问题总成立），$\text{rank}(A^HA) = \text{rank}(A) = n$ 可逆，若 $A = QR$ 为 thin QR，不难有 $\boldsymbol{x}^* = R^{-1}Q^H\boldsymbol{b}$ 为最小二乘问题的解，且该解（在列满秩时）唯一。

显然我们也可以通过直接使用 3.3 节的 Cholesky 算法解 $A^HA\boldsymbol{x}^* = A^H\boldsymbol{b}$ 从而得到一样的最小二乘问题的解。但这种方法 NOT backward stable。最后，我们也可以使用 QR 分解解方程 $A\boldsymbol{x} = \boldsymbol{b} \iff R\boldsymbol{x} = Q^H\boldsymbol{b}$，其中 QR 是 thin QR。这是稳定的，但注意如果使用 Householder $O(\frac{4}{3}n^3)$，其消耗的算力大概是 LU 的两倍 $O(\frac{2}{3}n^3)$。

5 Numerical Stability

5.1 Conditioning

Definition 5.1.1. 巴拿赫空间（Banach space） 若一个线性空间 X 上有 well-posed norm $||\cdot||_X$，则称 $(X\ ,\ ||\cdot||_X)$ 构成了一个 *Banach space*。

Definition 5.1.2. 绝对条件数（absolute condition number） 若 X, Y 为 Banach space，定义有一个函数 $f : U \subseteq X \mapsto X$ (s.t. $U \neq \emptyset$)，则其在 x 上的绝对条件数 *absolute condition number* 为

$$\hat{\kappa}_f(x) = \lim_{\varepsilon \to 0^+} \sup_{0<||\delta x||_X \ \leq \ \varepsilon} \frac{||\delta f||_Y}{||\delta x||_X}$$

其中 $\delta f = f(x+\delta x) - f(x)$，其衡量了在 x 处进行微小扰动 δx 时，函数值 $f(x)$ 的绝对误差之于 x 绝对误差的最大可能变化幅度，也即函数 f 在 x 邻域内的 *sensitivity*。

Definition 5.1.3. 相对条件数（relative condition number） 若 X, Y 为 Banach space，定义有一个函数 $f : U \subseteq X \mapsto Y$ (s.t. $U \neq \emptyset$)，则其在 x 上的相对条件数 *relative condition number* 为

$$\kappa_f(x) = \lim_{\varepsilon \to 0^+} \sup_{0<||\delta x||_X \ \leq \ \varepsilon} \frac{||\delta f||_Y}{||f(x)||_Y} \Big/ \frac{||\delta x||_X}{||x||_X} = \hat{\kappa}_f(x) \frac{||x||_X}{||f(x)||_Y}$$

其衡量了在 x 处进行微小扰动 δx 时，函数值 $f(x)$ 相对误差之于 x 相对误差的最大可能变化幅度。同样描述了函数 f 在 x 邻域内的 *sensitivity*。

Definition 5.1.4. 泛函导数 若 X, Y 为 Banach space，定义有一个函数 $f : U \subseteq X \mapsto Y$ (s.t. $U \neq \emptyset$)，若在 $x \in U$ 有

1) 任意方向 v 的方向导数 $D_v f(x) = \lim_{\varepsilon \to 0^+} \frac{f(x+\varepsilon v) - f(x)}{\varepsilon} \in Y$ 均存在，称为 *directionally differentiable*;

2) f directionally differentiable，且泛函 $Df(x)[\cdot] : v \in X \mapsto D_v f(x) \in Y$ 线性 + 有界，称为 *Gâteaux differentiable*。$Df(x)$ 为 *Gâteaux derivative*;

3) f Gâteaux differentiable，且 $\lim_{v \to 0} \frac{||f(x+v) - f(x) - Df(x)[v]||_Y}{||v||_X} = 0$ 对所有方向 $v \in X$ 均成立，称为 *Fréchet differentiable*。

Theorem 5.1.1. （绝对）条件数的计算
若 X, Y 为 Banach space，定义有一个函数 $f: U \subseteq X \mapsto Y$ (s.t. $U \neq \emptyset$)，若 f Fréchet differentiable，且有 Gâteaux derivative $Df(x)$，则 f 在 x 上的（绝对）条件数为

$$\hat{\kappa}_f(x) = ||Df(x)||_{XY} \quad \text{s.t.} \quad ||Df(x)||_{XY} = \sup_{h \in X \setminus \{0_X\}} \frac{||Df(x)h||_Y}{||h||_X}$$

证明见 NLA 补充材料 5.1 Proposition 7.5.3。该定理与证明均不作要求，只为辅助理解。

Definition 5.1.5. 矩阵条件数 在本课程中，我们主要关心的是可逆矩阵运算的条件数，其实际上是泛函 $f_A(x) = Ax$ 的全局相对条件数，即

$$\kappa(A) = \sup_x \kappa_{f_A}(x) = ||A||_2 \cdot ||A^{-1}||_2 = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

注意对于酉矩阵 $\kappa(Q) = 1$。

Proof Def 5.1.5 显然有 $\kappa_{f_A}(x) = \hat{\kappa}_{f_A}(x) \cdot \frac{||x||}{||f_A(x)||} = ||A||_2 \cdot \frac{||x||}{||Ax||}$，又有

$$\begin{aligned} \kappa_{f_A}(x) &= ||A||_2 \cdot \frac{||x||}{||Ax||} = ||A||_2 \cdot \frac{||A^{-1}Ax||}{||Ax||} \\ &\leq ||A||_2 \cdot ||A^{-1}||_2 = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)} \end{aligned}$$

因为总能取到 $x = v_{\cdot n}$ 为 $A \in \mathbb{F}^{n \times n}$ 的最后一个右奇异向量从而使得 $\frac{||x||}{||Ax||} = \frac{||v_{\cdot n}||}{||Av_{\cdot n}||} = \frac{1}{||\sigma_n u_{\cdot n}||} = \sigma_{\min}^{-1}(A)$，所以 $\sup_x \kappa_{f_A}(x) = ||A||_2 \cdot ||A^{-1}||_2 = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$。

Definition 5.1.6. Ill-conditioned Problem
若对于一个计算问题 evaluating $y = f(x)$ for given x，如果相对条件数 $\kappa_f(x) \gg 1$ (10^6 或更大)，则称这个问题是 *ill-conditioned* 的。其意味着即使是对自变量来说较小的（舍入）误差，在计算 $f(x)$ 时也会被大幅放大，从而导致计算精度的丢失。反之若 $\kappa_f(x) = O(1) \ll \epsilon_m^{-1}$（其中 ϵ_m 为机器精度，见下节）则称为 *well-conditioned*。注意 *conditioning* 是问题本身固有的性质，与所选算法无关，所以我们一般不会说一个算法是 *ill-conditioned* 的。

Theorem 5.1.2. Rule of Thumb
对于条件数的更“草率”的解释服从以下三条经验法则（*Rule of Thumb*）

1) 因变量绝对误差 $||\delta f|| \approx$ 绝对条件数 $\hat{\kappa}_f(x) \times$ 自变量绝对误差 $||\delta x||$
2) 因变量相对误差 $\frac{||\delta f||}{||f||} \approx$ 相对条件数 $\kappa_f(x) \times$ 自变量相对误差 $\frac{||\delta x||}{||x||}$
3) 若相对条件数 $\kappa_f(x) \approx 10^k$，则计算 $f(x)$ 大概会损失 k 个数位的精度

上述经验法则实际上还有更严谨的表述，特别地，我们有

Theorem 5.1.3. 误差的估计
若 $f(x)$ 的相对条件数为 $\kappa_f(x)$，则对于（充分小的）自变量相对误差 $E_r(x) := \frac{||\delta x||_X}{||x||_X}$（或者 δx），在 x 上总有

$$E_r(f) := \frac{||\delta f||_Y}{||f||_Y} = O(\kappa_f(x)E_r(x)) \quad \text{a.s.} \quad E_r(x) \to 0$$

同样地，若 $f(x)$ 的绝对条件数为 $\hat{\kappa}_f(x)$，则对于（充分小的）自变量绝对误差 $E(x) := ||\delta x||_X$（或者 δx），在 x 上总有

$$E(f) := ||\delta f||_Y = O(\hat{\kappa}_f(x)E(x)) \quad \text{a.s.} \quad E(x) \to 0$$

Proof Theorem 5.1.3 我们只对相对误差的情形予以证明，绝对误差的证明同理。由相对条件数与极限的定义

$$\kappa_f(x) = \lim_{\varepsilon \to 0^+} \sup_{0<||\delta x||_X \le \varepsilon} \frac{||\delta f||_Y}{||f(x)||_Y} / \frac{||\delta x||_X}{||x||_X} = \lim_{\varepsilon \to 0^+} \sup_{||\delta x||_X \in (0,\varepsilon)} \frac{E_r(f)}{E_r(x)}$$

我们有：任取 $C = \kappa_f(x)$，$\exists\ \delta_C > 0$ s.t. $\forall\ \varepsilon \in (0, \delta_C)$ 有

$$\left| \sup_{||\delta x||_X \in (0,\varepsilon)} \frac{E_r(f)}{E_r(x)} - k_f(x) \right| < C$$

不妨取 $\varepsilon = \frac{\delta_C}{2}$，则 $\forall\ 0 < E_r(x) < \frac{\delta_C}{2||x||_X} \Leftrightarrow 0 < ||\delta x||_X < \frac{\delta_C}{2} < \delta_C$ 有

$$\frac{E_r(f)}{E_r(x)} \le \sup_{||\delta x||_X \in (0, \frac{\delta_C}{2})} \frac{E_r(f)}{E_r(x)} < C + \kappa_f(x) = 2\kappa_f(x)$$

即有

$$\frac{E_r(f)}{E_r(x)} \le 2\kappa_f(x) \iff E_r(f) \le 2\kappa_f(x) E_r(x)$$

故，对 $\forall\ 0 < E_r(x) < \frac{\delta_C}{2||x||_X}$ 充分小

$$E_r(f) = O\left(\kappa_f(x) E_r(x)\right) \quad \text{a.s.} \quad E_r(x) \to 0$$

5.2 Floating Point Arithmetic and Stability

Definition 5.2.1. 浮点数集 由于计算机的存储容量是有限的，因此，在计算机中我们所能准确表达的数集 X 往往受到两条明显的限制：首先，X 只能是有限数集；其次，X 中的所有数都只能有有限位数。因此，要在计算机中表示实数集 $\mathbb{R}$，对实数进行舍入近似是不可避免的，因舍入产生的误差称为舍入误差 *rounding error*。在计算机中，（通过舍入）用以近似实数 $x \in \mathbb{R}$ 的数 $x' \in X$ 称为浮点数 *floating-point numbers*，我们用 $F := X$ 表示所有浮点数构成的集合。在计算机中，所有实数都用距离其最近的浮点数近似。

Definition 5.2.2. 机器精度 定义使用 $\hat{x} \in F$ 近似 $x \in \mathbb{R}$ 时所能达到的相对误差上限称为所用浮点数集的机器精度 *machine precision*，记作 ϵ_m 或 u (for *uite round-off*)，即

$$\forall\, x \in \mathbb{R} \;:\; \exists\, x' \in F \text{ s.t. } |x - x'| \le \epsilon_m |x|$$

对于目前最为普及的 64 位双精度浮点数 *IEEE 64-bit double-precision float number*，有 $\epsilon_m \approx 10^{-16}$。也即近似最多保证有 15 位有效数字 *significant number*。在不产生歧义的情况下，常记 $\epsilon = O(\epsilon_m) \iff |\epsilon| \le C\epsilon_m$ for $C \ll \epsilon_m^{-1}$。

Definition 5.2.3. 舍入误差与机器精度的估计
基于所选的 m bit 二进制浮点数类型，若其中有 n 位用于表示小数位（比如 64 位浮点数有 52 位用于表示小数数位），则

1） $\epsilon_m \approx \text{floor}\left[log_{10}(2^{n+1})\right]$

2） 计算最多保证有 $-log_{10}(\epsilon_m) - 1$ 位有效数字（若 relative error $< \frac{1}{2}10^{-k}$ 则估计值至少保留 k 位有效数字）

Definition 5.2.4. 浮点运算 接着 Def 5.2.2 我们定义浮点运算为映射 $fl(x) = x' \in F$，即

$$fl : \mathbb{R} \mapsto F \quad \text{s.t.} \quad fl(x) = x(1 + \epsilon_x) \quad \text{for some} \quad |\epsilon_x| \le \epsilon_m$$

对于浮点数运算，我们采用 IEEE 标准的运算法则

1） 对 $* := +/-/\times/\div$，$fl(x * y) = (x * y)(1 + \epsilon_*)$ s.t $|\epsilon_*| \le \epsilon_m$

2） 对 $* := +/\times$，$fl(x * y) = fl(y * x)$

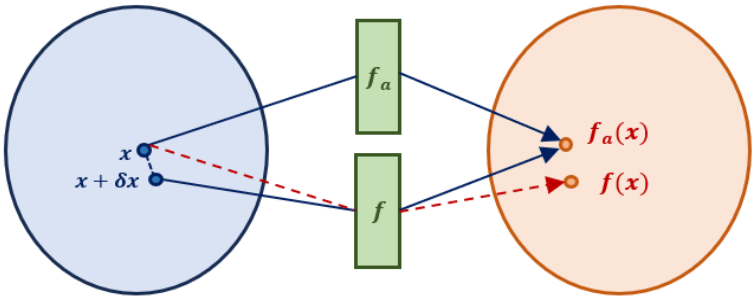
3） 对复合运算，$fl(\ (x *^1 y) *^2 z\ ) = fl(\ fl(x *^1 y)\ *^2\ z\ )$

注意根据 Def 5.2.2 自然有 $|fl(x) - x| \le \epsilon_m$ 和 $|fl(x * y) - x * y| \le \epsilon_m$，注意我们在后文中不会过多地讨论关于浮点运算误差的证明，必要时会直接给出结论。

Definition 5.2.5. Backward Stability
给定一个计算问题 evaluate $y = f(x)$ for given x precise, 简记 $p : y = f(x)$, 及一个用于估计 y 值的算法 $y \approx y_a = f_a(x)$ for given x precise。若

$$\forall\ x\ :\ \exists\ \delta x\ \text{ s.t. } E_r(x) = \frac{||\delta x||_X}{||x||_X} = \epsilon\ \text{ and }\ f_a(x) = y_a = f(x + \delta x)$$

则称该算法 *backward stable*，表示算法给出的解实际上等于某扰动问题下的精确解。其中 $\epsilon = O(\epsilon_m) \iff |\epsilon| \leq C\epsilon_m$ for $C \ll \epsilon_m^{-1}$。此处对于范数 $||\cdot||_X$ 一般是不重要的，可以按照需要进行选择和转换。这是因为可以证明在有限维线性空间 X 下，其上所有范数都是等价的（见补充材料 5.2.1）。



在证明 backward stability 时，一般有两种套路：一种可以直接尝试构造取 $\delta x = F(f_a(x), f(x), x)$ 来进行验证；另一种则是通过某种已知的 backward stability algorithm 进行转化。如果 $f(x), g(x)$ 是两个理论上的等价问题（解 y 相同），且知道其中之一的算法 $y_a = g_a(x)$ 为 backward stable，则可以由 $y_a = g(x + \delta x)$ 通过相应的转化实现 $g(x + \delta x) \Rightarrow f(x + \delta x')$ 并验证 $\delta x'$ 是否符合“足够小”的条件，特别是如果条件中有 $||\text{term}_1 - \text{term}_2||$ 的关系，则因考虑构造出 $g(x + \delta x) + \text{term}_1 - \text{term}_1 = f(x + \delta x')$（或者 $\pm\text{term}_2$），详细例子见 **thm 5.3.3 证明**。

Definition 5.2.6. Forward Stability
给定一个计算问题 $p : y = f(x)$，以及一个用于计算 y 估计值 $\hat{y}$ 的算法 $y \approx y_a = f_a(x)$ for given x precise。记 Banach Space Y 为值域，若

$$\forall\ x\ :\ E_r^a(f) := \frac{||f(x) - f_a(x)||_Y}{||f(x)||_Y} = O(k_f(x)\epsilon_m)$$

则称该算法 *forward stable*。

我们通常称上述定义中的精确自变量 x 为一个 problem，$y_a = f_a(x)$ 与 $y = f(x)$ 分别称为 computed/exact solution。如果一个算法是 backward stable，我们说 the algorithm offers an exact solution $f_a(x) = f(x + \delta x)$ for a slightly perturbed problem $x + \delta x$；如果算法是 forward stable，我们说 the

algorithm offers a perturbed solution $f_a(x)$ for a the problem x。特别的，如果 the algorithm offers a slightly perturbed solution $f_a(x)$ for a slightly perturbed problem $x+\delta x$，则称其为 mixed forward-backward stable。

我们知道，条件数能够将自变量和因变量关于函数 f（不是 f_a）的相对误差联系起来，这也是为什么我们需要引入 backward stable 这一概念的原因。因为其在 $f_a(x)$ 与 $f(x+\delta x)$ 之间建立了联系（从而 $E_r^a(f)=E_r(f)$ for some δx），使得我们可以通过对相应问题条件数的研究估计算法的稳定性。而 forward stable 的定义则不允许我们这么做。此外，实践证明好的算法往往都是 backward stable 的。

Definition 5.2.7. Accuracy
给定一个计算问题 $p: y=f(x)$，以及一个相应的算法 $y\approx y_a=f_a(x)$。记 Banach Space Y 为值域，若

$$\forall\ x\ :\ E_r^a(f):=\frac{||f(x)-f_a(x)||_Y}{||f(x)||_Y}=\epsilon=O(\epsilon_m)$$

则称该算法 *accurate*。注意，accurate 和 forward stable 的区别在于，如果一个算法是 forward stable 的，那么只有在 $\kappa_f(x)=O(1)\ll\epsilon_m^{-1}$ well-conditioned 时（一般取 1，10，100，1000 量级）才能说其是 accurate 的。

Theorem 5.2.1. Backward Stable + Well conditioned = Accurate
如果所给的算法 f_a 是 backward stable 的 $E_r(x)=\epsilon$ 则其一定 forward stable

$$E_r^a(f):=\frac{||f(x)-f_a(x)||_Y}{||f(x)||_Y}=O(\kappa_f(x)E_r(x))=O(\kappa_f(x)\epsilon_m)$$

$$E^a(f):=||f(x)-f_a(x)||_Y=O(\hat{\kappa}_f(x)E(x))=O(\hat{\kappa}_f(x)\epsilon_m)$$

有经验法则

$$E_r^a(f)\lesssim\kappa_f(x)E_r(x)\lesssim\kappa_f(x)\epsilon\ \ ;\ \ E^a(f)\lesssim\hat{\kappa}_f(x)E(x)\lesssim\hat{\kappa}_f(x)\epsilon$$

如果问题的条件数 $\kappa_f(x),\hat{\kappa}_f(x)=O(1)$ 足够小（well-conditioned），则所给算法 accurate。该定理还有一个矩阵版本，证明见 **Lecture Notes p.39 Theorem 7.1**。

Sketch Proof Theorem 5.2.1 由 Thm 5.1.3 可以有一些启发。对 $E_r(x)=\frac{||\delta x||_X}{||x||_X}\le C^*u$ s.t. $f_a(x)=f(x+\delta x)$，可取 $C>0$ s.t. $\delta_C>2C^*u||x||_X$，此时 $\forall\ 0<E_r(x)\le C^*u<\frac{\delta_C}{2||x||_X}$，有（绝对误差同理）

$$\frac{E_r^a(f)}{E_r(x)}=\frac{E_r(f)}{E_r(x)}<C+\kappa_f(x)\ \Rightarrow\ E_r^a(f)=O(\kappa_f(x)E_r(x))=O(\kappa_f(x)\epsilon_m)$$

5.3 Examples for Algorithm Stability

Definition 5.3.1. Some Basic Facts 在本节中，我们规定：记 $y = f(x)$ 为准确值，$f_a(x) := fl(f(x))$ 为相应的浮点“算法”，记估计值为 $y_a := \hat{y}$，有

1) 当 $n < \frac{1}{2\epsilon_m}$ 时，$f_a(x) = fl(x^T y)$ backward stable，有

$$fl(x^T y) = (x + \delta x')^T (y + \delta y') = (x + \delta x)^T y = x^T (y + \delta y)$$

其中 $||\delta x'||,\ ||\delta x|| = \epsilon||x||$, $||\delta y'||,\ ||\delta y|| = \epsilon||y||$ 感兴趣的可以查阅此链接或补充材料 5.3.1;

2) 由上一条事实自然有当 $n < \frac{1}{2\epsilon_m}$ 时（n 为矩阵列数），$f_a(x) = fl(Ax)$ backward stable，即

$$fl(Ax) = (A + \delta A)x \quad \text{for some} \quad ||\delta A|| \leq ||A||\epsilon$$

Theorem 5.3.1. 矩阵乘法的稳定性 浮点运算的矩阵乘法一般不是 backward stable 的，即一般 $fl(AB) \neq (A + \delta A)(B + \delta B)$；但是当 $n < \frac{1}{2\epsilon_m}$ 时（n 为中间维度）可以有

$$||fl(AB) - AB|| \leq \epsilon||A|| \cdot ||B||$$

进一步可以有相对误差限

$$\frac{||fl(AB) - AB||_2}{||AB||_2} \leq \epsilon \min\{\kappa(A), \kappa(B)\}$$

说明如果其中一个矩阵是准确输入，则矩阵浮点乘法 forward stable。

如果矩阵乘法中有一个是酉矩阵，则当中间维度 $n < \frac{1}{2\epsilon_m}$ 时，浮点运算的矩阵乘法 backward stable ，即有

$$fl(QB) = Q(B + \delta B) \quad \text{for} \quad \delta B = Q^H \left(fl(QB) - QB\right)$$

对右乘酉矩阵同理。

Theorem 5.3.2. 三角系统的稳定性 若考虑一个线性方程组 $R\boldsymbol{x} = \boldsymbol{b}$ 的求解问题，其中 R 是一个三角矩阵。则通过向后/前回代的方法求解该方程组是 backward stable 的，即（算法的解等于某个微扰问题的准确解）

$$\exists\ \delta R \text{ s.t. } ||\delta R||/||R|| = \epsilon \text{ and } (R + \delta R)\hat{\boldsymbol{x}} = b$$

特别地，对于方程组 $R_n \cdots R_1 \boldsymbol{x} = \boldsymbol{b}$，其中 R_i 均是三角矩阵，则有

$$\exists\ \delta R_i \text{ s.t. } ||\delta R_i||/||R_i|| = \epsilon \text{ and } (R_n + \delta R_n) \cdots (R_1 + \delta R_1)\hat{\boldsymbol{x}} = b$$

该定理的证明不做要求。

Proof Theorem 5.3.1 此处我们选取矩阵范数 $||A||_\infty = \max_{ij}|a_{ij}|$，由 $fl\left(x^T y\right) = x^T(y+\delta y)$ s.t. $||\delta y||_\infty \le ||y||_\infty C_n \epsilon_m$，则有

$$\begin{aligned}|fl(AB)-AB|_{ij} &= |fl(a_i^T b_j) - a_i^T b_j| = |a_i^T(b_j+\delta b_j) - a_i^T b_j| \\ &= |a_i^T \delta b_j| \le n||a_i||_\infty ||\delta b_j||_\infty \\ &\le nC_n\epsilon_m ||a_i||_\infty ||b_j||_\infty\end{aligned}$$

不妨记 $||fl(AB)-AB||_\infty = |fl(AB)-AB|_{pq}$，则

$$\begin{aligned}||fl(AB)-AB||_\infty &= |fl(AB)-AB|_{pq} \\ &\le nC_n\epsilon_m ||a_p||_\infty ||b_q||_\infty \\ &\le nC_n\epsilon_m ||A||_\infty ||B||_\infty = \epsilon ||A||_\infty ||B||_\infty\end{aligned}$$

由于同维矩阵范数是等价的，$||fl(AB)-AB|| \le \epsilon ||A||\cdot||B||$ 对所有矩阵范数均成立（可以直接证明对 $||\cdot||_F$ 也成立，见补充材料 5.3.2 p.5 Theorem 7.1 证法类似）。

由于上式对矩阵范数都成立，两边同时除以 $||AB||_2$，有（假设 A, B 均可逆）

$$\begin{aligned}\frac{||fl(AB)-AB||_2}{||AB||_2} &\le \epsilon \cdot \frac{||A||_2||B||_2}{||AB||_2} = \epsilon \cdot \frac{||A||_2||A^{-1}AB||_2}{||AB||_2} \\ &\le \epsilon \cdot \frac{||A||_2||A^{-1}||_2||AB||_2}{||AB||_2} = \epsilon \cdot ||A||_2||A_{-1}||_2 = \epsilon \cdot \kappa(A)\end{aligned}$$

同理，$\frac{||fl(AB)-AB||_2}{||AB||_2} \le \epsilon\cdot\kappa(B)$，于是 $\frac{||fl(AB)-AB||_2}{||AB||_2} \le \epsilon \cdot \min\{\kappa(A),\kappa(B)\}$。

下面讨论酉矩阵乘法的 backward stability：取 $\delta B = Q^H(fl(QB)-QB)$，不难证 $||\delta B||_2 = ||fl(QB)-QB||_2 \le \epsilon||Q||_2\cdot||B||_2 = \epsilon\cdot||B||_2 \Rightarrow \frac{||\delta B||_2}{||B||_2} = \epsilon = O(\epsilon_m)$。又有

$$f_Q(B+\delta B) = Q\left(B + Q^H(fl(QB)-QB)\right) = fl(QB)$$

即存在 $\delta B = Q^H(fl(QB)-QB)$ s.t. $\frac{||\delta B||_2}{||B||_2} = \epsilon = O(\epsilon_m)$，使得 $fl(QB) = f_Q(B+\delta B)$，有定义该浮点运算 backward stable。

Proof Theorem 5.3.2 简单证明一下推论，将方程组 $R_n\cdots R_1\boldsymbol{x} = \boldsymbol{b}$ 化为 $R_n y_n = \boldsymbol{b}$，$R_{n-1}y_{n-1} = y_n$，$\cdots$，$R_2 y_2 = y_3$，$R_1\boldsymbol{x} = y_2$ 分别运用稳定性结论即可：$(R_n+\delta R_n)\hat{y}_n = \boldsymbol{b}$，$(R_{n-1}+\delta R_{n+1})\hat{y}_{n-1} = \hat{y}_n$，$\cdots$，$(R_2+\delta R_2)\hat{y}_2 = \hat{y}_3$，$(R_1+\delta R_1)\hat{\boldsymbol{x}} = \hat{y}_2$，回代可得所需结论。

Theorem 5.3.3. LU 分解的稳定性
对于 LU 分解解方程组 $A\boldsymbol{x}=\boldsymbol{b}$，（在浮点运算的语境下）有以下结论

1） 不使用 Pivoting，分解得到的估值 $\hat{L}$，$\hat{U}$ 满足 $||\hat{L}\hat{U}-A||=\epsilon\,||L||\cdot||U||$；使用 Pivoting 后，满足 $||\hat{L}\hat{U}-PA||=\epsilon\,||\hat{L}||\cdot||\hat{U}||$

2） 当 $||L||\cdot||U||\,,\ ||\hat{L}||\cdot||\hat{U}||=O(||A||)$ 时，得到的解 $\hat{x}$ backward stable；

3） 使用 Pivoting 后，总有 $||\hat{L}||\,,\ ||L||=O(1)$，且当 $||\hat{U}||=O(||A||)$ 时，算法得到的解 $\hat{x}$ backward stable；

4） 当不使用 Pivoting 时，算法很容易不稳定（如 $\begin{bmatrix}\epsilon_m & 1\\ 1 & 1\end{bmatrix}$）；使用 Pivoting 后，在大部分的时候都相当稳定。

上述的 $\epsilon=O(\epsilon_m)$，且范数的选择不重要因为都等价。 第一条的结论证明不做要求，我们只证明 (2) (3) 并解释 (4) 中的反例，写出的证明比较重要，最好牢记流程。基本遵从了 def 5.2.5 的分析流程。

Proof Theorem 5.3.3 (2) 要证算法是 backward stable 的，我们需要证明 $(A+\delta A)\,\hat{x}=b$ 对解出的 $\hat{x}$ 和某个 $||\delta A||=\epsilon||A||$ 成立。首先我们的 $\hat{x}$（在没有 Pivoting 时）是通过解 $\hat{L}\hat{U}x=\boldsymbol{b}$ 解出的，由三角系统的稳定性，存在 $\delta\hat{L}$ 与 $\delta\hat{U}$ s.t.

$$(\hat{L}+\delta\hat{L})(\hat{U}+\delta\hat{U})\hat{x}=\boldsymbol{b}\ \text{ s.t. }\ ||\delta\hat{L}||=\epsilon||\hat{L}||\ \text{and}\ ||\delta\hat{U}||=\epsilon||\hat{U}||$$

由于有 $||\hat{L}\hat{U}-A||=\epsilon\,||L||\cdot||U||$，我们可以构造

$$\begin{aligned}\left(\hat{L}+\delta\hat{L}\right)\left(\hat{U}+\delta\hat{U}\right)\hat{x}&=\left(\hat{L}\hat{U}+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\hat{x}\\&=\left(A+\hat{L}\hat{U}-A+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\hat{x}\\&=\left[A+\left(\hat{L}\hat{U}-A+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\right]\hat{x}=\boldsymbol{b}\end{aligned}$$

其中 $\delta A=\hat{L}\hat{U}-A+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}$，下面检验其大小

$$\begin{aligned}||\delta A||&=||\hat{L}\hat{U}-A+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}||\\&\le||\hat{L}\hat{U}-A||+||\delta\hat{L}||\cdot||\hat{U}||+||\hat{L}||\cdot||\delta\hat{U}||+||\delta\hat{L}||\cdot||\hat{U}||\\&=\epsilon||L||\cdot||U||+\epsilon||\hat{L}||\cdot||\hat{U}||+\epsilon||\hat{L}||\cdot||\hat{U}||+\epsilon^2||\hat{L}||\cdot||\hat{U}||\\&=\epsilon||L||\cdot||U||+\epsilon||\hat{L}||\cdot||\hat{U}||=\epsilon O(||A||)\ \Rightarrow\ ||\delta A||/||A||=\epsilon\end{aligned}$$

(3) 证明基本类似，我们的 $\hat{x}$ （在 Pivoting 时）是通过解 $\hat{L}\hat{U}x = P\boldsymbol{b}$ 解出的，由三角系统的稳定性，存在 $\delta\hat{L}$ 与 $\delta\hat{U}$ s.t.

$$(\hat{L}+\delta\hat{L})(\hat{U}+\delta\hat{U})\hat{x} = P\boldsymbol{b} \text{ s.t. } ||\delta\hat{L}|| = \epsilon||\hat{L}|| \text{ and } ||\delta\hat{U}|| = \epsilon||\hat{U}||$$

由于有 $||\hat{L}\hat{U} - PA|| = \epsilon\,||\hat{L}||\cdot||\hat{U}||$ ，我们可以构造

$$\begin{aligned}\left(\hat{L}+\delta\hat{L}\right)\left(\hat{U}+\delta\hat{U}\right)\hat{x} &= \left(\hat{L}\hat{U}+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\hat{x}\\ &= \left(PA+\hat{L}\hat{U}-PA+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\hat{x}\\ &= P\left[A+P^T\left(\hat{L}\hat{U}-PA+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\right]\hat{x} = P\boldsymbol{b}\\ &\Rightarrow \left[A+P^T\left(\hat{L}\hat{U}-PA+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)\right]\hat{x} = \boldsymbol{b}\end{aligned}$$

其中 $\delta A = P^T\left(\hat{L}\hat{U}-PA+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}\right)$，下面检验其大小，首先由于 P^T 为酉矩阵，我们考虑酉不变范数 $||\cdot||_2$

$$\begin{aligned}||\delta A||_2 &= ||\hat{L}\hat{U}-PA+\delta\hat{L}\hat{U}+\hat{L}\delta\hat{U}+\delta\hat{L}\delta\hat{U}||_2\\ &\le ||\hat{L}\hat{U}-PA||_2 + ||\delta\hat{L}||_2||\hat{U}||_2 + ||\hat{L}||_2||\delta\hat{U}||_2 + ||\delta\hat{L}||_2||\hat{U}||_2\\ &= \epsilon||\hat{L}||_2||\hat{U}||_2 + \epsilon||\hat{L}||_2||\hat{U}||_2 + \epsilon||\hat{L}||_2||\hat{U}||_2 + \epsilon^2||\hat{L}||_2||\hat{U}||_2\\ &= \epsilon||\hat{L}||_2\cdot||\hat{U}||_2 = \epsilon\cdot O(1)\cdot||\hat{U}||_2 = \epsilon O(||A||_2) \;\Rightarrow\; ||\delta A||_2/||A||_2 = \epsilon\end{aligned}$$

注意此处我们利用了：使用 Pivoting 后，总有 $||\hat{L}||\,,\ ||L|| = O(1)$，这是因为 PLU 算法确保了 $L,\hat{L}$ 的每一个元素都小于（相差机器精度），从而 $||L||_\infty\,,\ ||\hat{L}||_\infty = O(1)$，结合矩阵范数的等价性易得该结论。此外，在这里的 ϵ 实际上都指代 $O(\epsilon_m)$ 而不是一个具体的数，所以可以有类似 $\epsilon+\epsilon=\epsilon$ 的推断出现。

(4) 我们考察矩阵范数 $||\cdot||_\infty$，对于 LU 分解

$$A = \begin{bmatrix}\epsilon_m & 1\\ 1 & 1\end{bmatrix} = \begin{bmatrix}1 & 0\\ \epsilon_m^{-1} & 1\end{bmatrix}\begin{bmatrix}\epsilon_m & 1\\ 0 & 1-\epsilon_m^{-1}\end{bmatrix} = LU$$

显然有 $||L||_\infty||U||_\infty = |\epsilon_m^{-1}|\cdot|1-\epsilon_m^{-1}| \gg ||A||_\infty = 1$，不满足第二条的条件。

Theorem 5.3.4. 一些零散结论

1) 解正定对称方程组的 Cholesky 算法 backward stable，且无需 Pivoting；

2) Orthogonal triangularisation 的 QR 算法一般是 backward stable 的，因此用它们进行 QR 分解解方程组的算法也是 backward stable 的；

3) (这一条不要求掌握) CGS 与 MGS 都不是 backward stable 的。特别地，$||\hat{Q}^T\hat{Q} - I||$ 对 CGS，MGS 分别有，$O(\epsilon(\kappa(A))^2)$ 与 $O(\epsilon\kappa(A))$。有 Gram-Schmidt Twice 算法可以达到 $O(\epsilon)$。

4) Householder QR 是 backward stable 的，即对于计算出的结果 $\hat{Q}$，$\hat{R}$，满足

$$||\hat{Q}^T\hat{Q} - I|| = O(\epsilon) \quad \text{and} \quad ||A - \hat{Q}\hat{R}|| = O(\epsilon||A||)$$