Exercise 2 (Eigenvalues of a Rank-1 Matrix): (*15* points)
Let $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ be a real, symmetric matrix with rank one.

a) Show that $\boldsymbol{A}=\boldsymbol{u}\boldsymbol{u}^\top$ for some nonzero real vector $\boldsymbol{u}\in\mathbb{R}^n$.

b) Show that $\|\boldsymbol{u}\|^2$ is an eigenvalue of $\boldsymbol{A}$.

c) What are the other eigenvalues of $\boldsymbol{A}$?

d) If the power iteration is applied to $\boldsymbol{A}$ with initial point $\boldsymbol{x}^0\in\mathbb{R}^n$ satisfying $\boldsymbol{u}^\top\boldsymbol{x}^0\neq 0$, how many iterations are required for it to converge *exactly* to the eigenvector corresponding to the dominant eigenvalue? Explain your answer!

Solution :

a) If A is rank one, then the dimension of the linear space spanned by the columns of A is 1. Assume $\{\boldsymbol{w}\}$ is a basis of the column space of $\boldsymbol{A}=(\boldsymbol{a}_1\ \boldsymbol{a}_2\cdots\boldsymbol{a}_n)$. Then there exists $\boldsymbol{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_n)^T\in\mathbb{R}^n$ such that $\boldsymbol{a}_i=\alpha_i\boldsymbol{w}$ for all $i=1,2,\ldots,n$, i.e.

$$\boldsymbol{A}=\boldsymbol{w}\boldsymbol{\alpha}^T \tag{6}$$

Since $\boldsymbol{A}=\boldsymbol{A}^T=\boldsymbol{\alpha}\boldsymbol{w}^T$, then $\boldsymbol{\alpha}\in\text{span}\{\boldsymbol{w}\}$. There exists a nonzero real value β such that $\boldsymbol{\alpha}=\beta\boldsymbol{w}$, which implies $\boldsymbol{A}=\beta\boldsymbol{w}\boldsymbol{w}^T$. Since $\boldsymbol{A}$ is positive semidefinite, we have

$$\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w}=\beta\|\boldsymbol{w}\|^4\geqslant 0, \tag{7}$$

which implies $\beta>0$. Denote $\boldsymbol{u}=\sqrt{\beta}\boldsymbol{w}$, we have

$$\boldsymbol{A}=\boldsymbol{u}\boldsymbol{u}^T. \tag{8}$$

b) Notice that $\boldsymbol{A}\boldsymbol{u}=\|\boldsymbol{u}\|^2\boldsymbol{u}$, thus $\|\boldsymbol{u}\|^2$ is an eigenvalue of $\boldsymbol{A}$.

c) Notice that for any $\boldsymbol{v}\perp\boldsymbol{u}$ and $\boldsymbol{v}\neq 0$, we have

$$\boldsymbol{A}\boldsymbol{v}=(\boldsymbol{u}^T\boldsymbol{v})\boldsymbol{u}=0, \tag{9}$$

which implies $(\text{span}\{\boldsymbol{u}\})^\perp$ is the eigenspace corresponding to eigenvalue 0. Since $\text{span}\{\boldsymbol{u}\}\oplus(\text{span}\{\boldsymbol{u}\})^\perp=\mathbb{R}^n$, we can conclude that the only other eigenvalue is 0.

d) Only 1 step. Just notice that $\boldsymbol{A}\boldsymbol{x}^0=(\boldsymbol{u}^T\boldsymbol{x}^0)\boldsymbol{u}$, thus $\boldsymbol{x}^1=\frac{\boldsymbol{A}\boldsymbol{x}^0}{\|\boldsymbol{A}\boldsymbol{x}^0\|}=\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}$ which is just the the eigenvector corresponding to the dominant eigenvalue $\|\boldsymbol{u}\|^2$.

Exercise 5 (Singular Value Decomposition): (*20* points)
In this problem, we consider the SVD decomposition of a matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$ given its low-rank structure $\boldsymbol{A}=\boldsymbol{B}\boldsymbol{C}^\top$, where $\boldsymbol{B},\boldsymbol{C}\in\mathbb{R}^{n\times k}$ are two known full rank matrices with $n>k$.

a) Design an algorithm to compute the SVD of $\boldsymbol{A}$ based on the given matrices $\boldsymbol{B}$ and $\boldsymbol{C}$, i.e., without forming the matrix $\boldsymbol{A}$ explicitly.

– Your algorithm can use QR factorizations and SVDs (of matrices different from $\boldsymbol{A}$).

– Suppose we only consider the computational costs of the involved matrix decompositions (QR factorizations, SVDs, etc.). Is your algorithm more efficient compared to applying the SVD directly to $\boldsymbol{A}$? Explain your answer!

Hint: You can assume that the computational costs of the QR factorization and SVD of a general $m\times p$ matrix with $m\geq p$ are given by $\mathcal{O}(mp^2)$ and $\mathcal{O}(m^2p)$, respectively.

Solution :

a) The procedure is as follows:

- Find the reduced QR factorization of $\boldsymbol{B}$ and $\boldsymbol{C}$: $\boldsymbol{B}=\boldsymbol{Q_B}\boldsymbol{R_B},\boldsymbol{C}=\boldsymbol{Q_C}\boldsymbol{R_C}$, where $\boldsymbol{Q_B},\boldsymbol{Q_C}\in\mathbb{R}^{n\times k}$ and $\boldsymbol{R_B},\boldsymbol{R_C}\in\mathbb{R}^{k\times k}$.
- Find the SVD decomposition of the k by k matrix $\boldsymbol{R_B}\boldsymbol{R_C}^\top$: $\boldsymbol{R_B}\boldsymbol{R_C}^\top=\bar{\boldsymbol{U}}\boldsymbol{\Sigma}\bar{\boldsymbol{V}}^\top$, where $\bar{\boldsymbol{U}},\boldsymbol{\Sigma},\bar{\boldsymbol{V}}\in\mathbb{R}^{k\times k}$.
- Let $\boldsymbol{U}=\boldsymbol{Q_B}\bar{\boldsymbol{U}}$ and $\boldsymbol{V}=\boldsymbol{Q_C}\bar{\boldsymbol{V}}$. Then $\boldsymbol{A}=\boldsymbol{B}\boldsymbol{C}^\top=\boldsymbol{Q_B}\boldsymbol{R_B}\boldsymbol{R_C}^\top\boldsymbol{Q_C}^\top=\boldsymbol{Q_B}\bar{\boldsymbol{U}}\boldsymbol{\Sigma}\bar{\boldsymbol{V}}^\top\boldsymbol{Q_C}^\top=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the SVD decomposition of $\boldsymbol{A}$.

Since this algorithm requires the QR decomposition of two n by k matrix and the SVD decomposition of a k by k matrix, the total cost of it is $2\mathcal{O}(nk^2)+\mathcal{O}(k^3)=\mathcal{O}(nk^2)$. Note that the computation cost of the direct SVD decomposition of $\boldsymbol{A}$ is $\mathcal{O}(n^3)$. The new algorithm is more efficient when $k\ll n$.

Exercise 4 (The Conjugate Gradient (CG) Method): (*23* points)
We consider the linear equation $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$, where

$$\boldsymbol{A}=\begin{bmatrix}2&-1\\1&-2\end{bmatrix}\quad\text{and}\quad\boldsymbol{b}=\begin{bmatrix}0\\-3\end{bmatrix}.$$

a) Show that $\boldsymbol{x}$ is a solution to $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ if and only if $\boldsymbol{x}$ solves the modified linear system

$$\boldsymbol{D}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{D}\boldsymbol{b}\quad\text{where}\quad\boldsymbol{D}=\begin{bmatrix}1&0\\0&-1\end{bmatrix}.$$

b) Can the modified system $\boldsymbol{D}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{D}\boldsymbol{b}$ be solved via CG? If yes, apply the CG method with initial point $\boldsymbol{x}^0=\boldsymbol{0}$ and compute the respective solution $\boldsymbol{x}$ to the linear equation $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$.

c) Consider a general linear system $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ with symmetric, positive definite matrix $\boldsymbol{A}\in\mathbb{R}^{n\times n}$. Suppose $\boldsymbol{b}\in\mathbb{R}^n\backslash\{0\}$ is an eigenvector of $\boldsymbol{A}$ and let us apply the conjugate gradient method to solve $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ with initial point $\boldsymbol{x}^0=\boldsymbol{0}$. Perform one step of the CG method – what can you say about the iterate $\boldsymbol{x}^1$?

Solution :

a) The matrix $\boldsymbol{D}$ is nonsingular. Hence, multiplying the system $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ with $\boldsymbol{D}$ (from the left) does not change its solution.

b) The CG method requires the system matrix to be symmetric and positive definite. Here, we have

$$\tilde{\boldsymbol{A}}:=\boldsymbol{D}\boldsymbol{A}=\begin{bmatrix}1&0\\0&-1\end{bmatrix}\begin{bmatrix}2&-1\\1&-2\end{bmatrix}=\begin{bmatrix}2&-1\\-1&2\end{bmatrix}\quad\text{and}\quad\tilde{\boldsymbol{b}}:=\boldsymbol{D}\boldsymbol{b}=\begin{bmatrix}0\\3\end{bmatrix}.$$

$\boldsymbol{D}\boldsymbol{A}$ is obviously symmetric and it holds that $\text{tr}(\boldsymbol{D}\boldsymbol{A})=4>0$ and $\det(\boldsymbol{D}\boldsymbol{A})=4-1=3>0$. Hence, all eigenvalues of $\boldsymbol{D}\boldsymbol{A}$ are positive establishing positive definiteness.

Hence, the CG-method is applicable. Since we are solving a two dimensional system, we only need to perform two CG steps. Setting $\boldsymbol{x}^0=\boldsymbol{0}$, it follows $\boldsymbol{r}^0=\tilde{\boldsymbol{A}}\boldsymbol{x}^0-\tilde{\boldsymbol{b}}=[0,-3]^\top$, $\boldsymbol{p}^0=-\boldsymbol{r}^0=[0,3]^\top$.

Step 1: $\alpha_0=\frac{\|\boldsymbol{r}^0\|^2}{(\boldsymbol{p}^0)^\top\tilde{\boldsymbol{A}}\boldsymbol{p}^0}=\frac{9}{2\cdot 9}=\frac12$, $\boldsymbol{x}^1=\boldsymbol{x}^0+\alpha_0\boldsymbol{p}^0=[0,\frac32]^\top$, $\boldsymbol{r}^1=\boldsymbol{r}^0+\alpha_0\tilde{\boldsymbol{A}}\boldsymbol{p}^0=[-\frac32,0]^\top$, $\beta=\frac{\|\boldsymbol{r}^1\|^2}{\|\boldsymbol{r}^0\|^2}=\frac14$, $\boldsymbol{p}^1=-\boldsymbol{r}^1+\beta\boldsymbol{p}^0=[\frac32,\frac34]^\top$.

Step 2: $\alpha_1=\frac{\|\boldsymbol{r}^1\|^2}{(\boldsymbol{p}^1)^\top\tilde{\boldsymbol{A}}\boldsymbol{p}^1}=\frac{9/4}{9/4\cdot 3/2}=\frac23$, $\boldsymbol{x}^2=\boldsymbol{x}^1+\alpha_1\boldsymbol{p}^1=[1,2]^\top$.

c) Suppose $\boldsymbol{A}\boldsymbol{b}=\lambda\boldsymbol{b}$, where λ is the eigenvalue corresponding to $\boldsymbol{b}$. Then $\boldsymbol{r}^0=\boldsymbol{A}\boldsymbol{x}^0-\boldsymbol{b}=-\boldsymbol{b}$ and $\boldsymbol{p}^0=-\boldsymbol{r}^0=\boldsymbol{b}$. Moreover,

$$\alpha_0=\frac{\|\boldsymbol{r}^0\|^2}{(\boldsymbol{p}^0)^\top\boldsymbol{A}\boldsymbol{p}^0}=\frac{\|\boldsymbol{b}\|^2}{\boldsymbol{b}^\top(\boldsymbol{A}\boldsymbol{b})}=\frac{\|\boldsymbol{b}\|^2}{\boldsymbol{b}^\top(\lambda\boldsymbol{b})}=\frac{1}{\lambda}.$$

So $\boldsymbol{x}^1=\boldsymbol{x}^0+\alpha_0\boldsymbol{p}^0=\boldsymbol{b}/\lambda$ and $\boldsymbol{r}^1=\boldsymbol{r}^0+\alpha_0\boldsymbol{A}\boldsymbol{p}^0=-\boldsymbol{b}+\alpha_0\boldsymbol{A}\boldsymbol{b}=\boldsymbol{0}$. Thus, $\boldsymbol{x}^1$ is the solution to $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$.

注：可通过因式分解、分子有理化、恒等式、泰勒展开等方法处理两数相减

① 二倍角公式：$\cos(2x)=1-2\sin^2x$，$\sin(2x)=2\sin x\cos x$

② Taylor expansion:

① $e^x=1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\cdots+\frac{x^n}{n!}+\cdots\ (-\infty<x<+\infty)$

② $\sin x=x-\frac{x^3}{3!}+\frac{x^5}{5!}-\cdots+(-1)^n\frac{x^{2n+1}}{(2n+1)!}+\cdots\ (-\infty<x<+\infty)$

③ $\frac{1}{1-x}=1+x+x^2+x^3+\cdots+x^n+\cdots\ (-1<x<1)$

④ $\ln(1+x)=x-\frac{x^2}{2}+\frac{x^3}{3}-\frac{x^4}{4}+\cdots+(-1)^{n-1}\frac{x^n}{n}+\cdots\ (-1<x\leq 1)$

⑤ $\cos x=1-\frac{x^2}{2!}+\frac{x^4}{4!}-\frac{x^6}{6!}+\cdots+(-1)^n\frac{x^{2n}}{(2n)!}+\cdots\ (-\infty<x<+\infty)$

⑥ $a^x=1+x\ln a+\frac{x^2(\ln a)^2}{2!}+\cdots+\frac{x^n(\ln a)^n}{n!}+\cdots\ (-\infty<x<+\infty)$

⑦ $\frac{1}{1+x^2}=1-x^2+x^4-x^6+\cdots+(-1)^nx^{2n}+\cdots\ (-1<x<1)$

⑧ $\arctan x=x-\frac{x^3}{3}+\frac{x^5}{5}-\frac{x^7}{7}+\cdots+(-1)^n\frac{x^{2n+1}}{2n+1}\ (-1\leq x\leq 1)$

---

③ Backward error：$\Delta x=\hat{x}-x=f^{-1}(\hat{f}(x))-x$

能通过原函数得出 $\hat{f}$ 的 input $\hat{x}$ ($f(\hat{x})=\hat{y}$) 与 actual input x 的差值

④ Relative backward error：$\frac{|\Delta x|}{|x|}=\frac{|\hat{x}-x|}{|x|}=\frac{|f^{-1}(\hat{f}(x))-x|}{|x|}$

§3 Conditioning

1. well-posed problem：solution ① exists ② unique ③ depends continuously on problem data

2. sensitivity 与 conditioning 一个 problem 被称为

① insensitive/well-conditioned：input 的 relative change ≈ solution 的 relative change

② sensitive/ill-conditioned：input 的 relative change ≪ solution 的 relative change

3. Condition number (条件数)

$$\text{cond}=\frac{|\text{relative change in solution}|}{|\text{relative change in input data}|}=\frac{|[f(\hat{x})-f(x)]/f(x)|}{|(\hat{x}-x)/x|}=\frac{|\Delta y/y|}{|\Delta x/x|}$$

注：cond ≤ $10^3/10^4$ ⇒ well-conditioned；cond ≥ $10^6/10^8$ ⇒ ill-conditioned

表示了 Relative forward error 相对于 Relative backward error 的 amplification factor

实际中通常会采 |Relative forward error| ≲ cond · |Relative backward error|

4. 条件数的计算：w/ evaluating a function 为例

Condition number $\text{cond}\approx\left|\frac{f(x+\Delta x)/f(x)}{\Delta x/x}\right|=\left|\frac{xf'(x)}{f(x)}\right|$

注：在上例中，有 $\text{cond}(f^{-1})=\frac{1}{\text{cond}(f)}$

证 $\frac{1}{\text{cond}(f(x))}=\left|\frac{f(x)}{xf'(x)}\right|$　$\text{cond}(f^{-1}(y))=\left|\frac{y\,(f^{-1})'(y)}{f^{-1}(y)}\right|$　令 $y=f(x)$

§4 Stability

1. Stability：一个 problem $f:R^n\to R^m$ 的 algorithm $\hat{f}:R^n\to R^m$ 被称为 stable，若对 $\forall x\in R^n$，有 $\frac{\|\hat{f}(x)-f(\hat{x})\|}{\|f(\hat{x})\|}=O(\varepsilon_{mach})$ for some $\hat{x}$：$\frac{\|\hat{x}-x\|}{\|x\|}=O(\varepsilon_{mach})$

对所有 input x，algorithm $\hat{f}(x)$ 的结果都近似等于(某)近似问题($f(\hat{x})$)的解

2. big-O：$f(x)=O(g(x))\iff\exists\,\varepsilon>0,C>0$ s.t. $|f(x)|\leq C\,g(x)$，$\forall\,|x-a|<\varepsilon$

3. small-o：$f(x)=o(g(x))$ for $x\to a\iff\lim_{x\to a}|f(x)|/|g(x)|=0$

§5 Accuracy

1. Accuracy 一个 problem $f:R^n\to R^m$ 的 algorithm $\hat{f}:R^n\to R^m$ 被称为 accurate，若对 $\forall x\in R^n$，有 $\frac{\|\hat{f}(x)-f(x)\|}{\|f(x)\|}=O(\varepsilon_{mach})$ (for $\varepsilon_{mach}\to 0$)

所有 input data 经 algorithm 处理后得到的结果都接近原问题的结果

2. Theorem：若 ① $f:R^n\to R^m$ 的 relative condition number 为 cond

② $\hat{f}$ 为 stable algorithm，则 $\frac{\|\hat{f}(x)-f(x)\|}{\|f(x)\|}\approx O(\text{cond}(x)\cdot\varepsilon_{mach})$

Exercise 1 (Inverse Iteration): (*16* points)
We apply the inverse iteration

$$(\boldsymbol{A}-\sigma\boldsymbol{I})\boldsymbol{x}=\boldsymbol{x}^{k-1},\quad \boldsymbol{x}^k=\boldsymbol{x}/\|\boldsymbol{x}\|,\quad \sigma_k=(\boldsymbol{x}^k)^\top\boldsymbol{A}\boldsymbol{x}^k$$

with fixed shift $\sigma=10$ and initial point $\boldsymbol{x}^0=[1,1,1,1]^\top/\sqrt{4}$ to the matrices

$$\boldsymbol{A}_1=\begin{bmatrix}-5&0&0&0\\0&-1&0&0\\0&0&8&0\\0&0&0&11\end{bmatrix},\quad \boldsymbol{A}_2=\begin{bmatrix}-2&0&0&1\\0&-1&0&0\\0&0&1&0\\0&0&0&11\end{bmatrix},\quad\text{and}\quad \boldsymbol{A}_3=\begin{bmatrix}-2&0&0&1\\0&-1&0&0\\0&0&9&0\\0&0&0&11\end{bmatrix}.$$

a) The plot below shows the error $|\sigma_k-11|$ (on the y-axis, using a logarithmic scale) versus the number of iterations k for the inverse iteration applied to the matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, and $\boldsymbol{A}_3$.

Match each of the plotted lines (circle: ●, square: ■, diamond: ◆) to one of the respective matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, or $\boldsymbol{A}_3$ and provide a detailed explanation of your matching.

b) Compute the slope of the middle line (square black markers). Notice again that the plot in part a) is a logarithmic plot depicting $\log_{10}(|\sigma_k-11|)$ on the y-axis.

Remark: The shown plot cannot be used directly to "graphically" measure the slope of the line.

Solution :

a) The matching is given by [●; $\boldsymbol{A}_3$], [■; $\boldsymbol{A}_1$], [◆; $\boldsymbol{A}_2$].

We need to consider the eigenvalues of $(\boldsymbol{A}_i-10\boldsymbol{I})^{-1}$. The matrix $(\boldsymbol{A}_3-10\boldsymbol{I})^{-1}$ does not have a dominant eigenvalue and hence, we cannot expect convergence of the shifted inverse iteration applied to $\boldsymbol{A}_3$. The eigenvalues of $(\boldsymbol{A}_1-10\boldsymbol{I})^{-1}$ are given by

$$\frac{1}{|11-10|},\ \frac{1}{|8-10|},\ \frac{1}{|-1-10|},\ \frac{1}{|-5-10|}\quad\to\quad 1,\ \frac12,\ \frac{1}{11},\ \frac{1}{15}.$$

Hence, since $\boldsymbol{A}_1$ is symmetric, we expect the convergence rate $(\frac12/1)^2=\frac14$. The eigenvalues of $(\boldsymbol{A}_2-10\boldsymbol{I})^{-1}$ are given by

$$\frac{1}{|11-10|},\ \frac{1}{|1-10|},\ \frac{1}{|-1-10|},\ \frac{1}{|-2-10|}\quad\to\quad 1,\ \frac19,\ \frac{1}{11},\ \frac{1}{12}.$$

The convergence rate in this case is $\frac19/1=\frac19$. Consequently, convergence for $\boldsymbol{A}_2$ is faster than for $\boldsymbol{A}_1$.

b) The plot depicts the error $|\sigma_k-11|\approx c|\lambda_2/\lambda_1|^{2k}$ where λ_1 is the largest and λ_2 is the second largest eigenvalue of $(\boldsymbol{A}_1-10\boldsymbol{I})^{-1}$ (after taking absolute values and for some suitable $c>0$). Hence, as shown before the ratio $|\lambda_2/\lambda_1|$ is exactly $\frac12$ and we have

$$\log_{10}(|\sigma_k-11|)=\log_{10}(c)+k\cdot 2\log_{10}(1/2).$$

Consequently, the slope is given by $-2\log_{10}(2)$.

---

3. Systems of Linear Equations

§1 Vector norm

① $\|x\|_1\geq\|x\|_2\geq\|x\|_\infty$　② $\|x\|_1\leq\sqrt{n}\cdot\|x\|_2$

③ $\|x\|_2\leq\sqrt{n}\cdot\|x\|_\infty$　④ $\|x\|_1=\sum_{i=1}^n|x_i|\leq\sum_{i=1}^n\max|x_i|=n\cdot\|x\|_\infty$

§2 Matrix norms

1. Matrix norms 的基本要求

① Positivity：$\|A\|>0$ 若 $A\neq 0$；$\|A\|=0$ 当且仅当 $A=0$

② Positive homogeneity：$\|\alpha A\|=|\alpha|\cdot\|A\|$ for $\forall$ scalar $\alpha\in R$

③ Triangle inequality：$\|A+B\|\leq\|A\|+\|B\|$，for $\forall$ A, B

(vector norm 的性质)

④ Sub-multiplicative：$\|AB\|\leq\|A\|\cdot\|B\|$，for $\forall$ A, B (CS inequality 的延伸)

⑤ 与 vector norm compatible：$\|Ax\|\leq\|A\|\cdot\|x\|$ for $\forall$ A, vector x

2. Induced matrix norm (表示在给定 vector norm 下，A 能造成多大的相对拉伸)

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

3. 常见的 induced matrix norms

① l_1 norm (列和范数) $\|A\|_1:=\max_{j\in\{1,\cdots,n\}}\sum_{i=1}^m|a_{ij}|$

② l_2 norm/spectual norm (谱范数) $\|A\|_2:=\sqrt{\lambda_{max}(A^TA)}$

③ ∞-norm (行和范数) $\|A\|_\infty:=\max_{i\in\{1,\cdots,m\}}\sum_{j=1}^n|a_{ij}|$

证 Derivation for l_1 matrix norm

定义 $A_{m\times n}=\begin{bmatrix}a_1^T\\a_2^T\\\vdots\\a_m^T\end{bmatrix}=[A_1,A_2,\cdots,A_n]$ 有 $\|A_j\|_1=\sum_{i=1}^m|a_{ij}|$

① 证明 $\frac{\|Ax\|_1}{\|x\|_1}\leq\max_{j\in\{1,\cdots,n\}}\sum_{i=1}^m|a_{ij}|$

$$\|Ax\|_1=\left\|\begin{bmatrix}a_1^Tx\\\vdots\\a_m^Tx\end{bmatrix}\right\|_1=\sum_{i=1}^m|a_i^Tx|=\sum_{i=1}^m\left|\sum_{j=1}^na_{ij}\cdot x_j\right|\leq\sum_{i=1}^m\sum_{j=1}^n|a_{ij}|\cdot|x_j|$$

$$=\sum_{j=1}^n\sum_{i=1}^m|a_{ij}|\cdot|x_j|=\sum_{j=1}^n\|A_j\|_1\cdot|x_j|\leq\max_{j\in\{1,\cdots,n\}}\|A_j\|_1\cdot\sum_{j=1}^n|x_j|=\max_{j\in\{1,\cdots,n\}}\|A_j\|_1\cdot\|x\|_1$$

$$\Rightarrow\frac{\|Ax\|_1}{\|x\|_1}\leq\max_{j\in\{1,\cdots,n\}}\|A_j\|_1=\max_{j\in\{1,\cdots,n\}}\sum_{i=1}^m|a_{ij}|$$

② 证明 $\exists\,x$，s.t. $\frac{\|Ax\|_1}{\|x\|_1}=\max_{j\in\{1,\cdots,n\}}\sum_{i=1}^m|a_{ij}|$

若 $\max_{j\in\{1,\cdots,n\}}\sum_{i=1}^m|a_{ij}|=\sum_{i=1}^m|a_{ij_0}|$，则选 x 为 unit vector，且 $x_{j_0}=1$

证 Derivation for l_2 matrix norm

令 $(\lambda_i,u_i)_{i=1}^n$ 表示 A^TA 的 eigenvalue-eigenvector pairs，有

① $A^TAu_i=\lambda_iu_i$　② $u_i^Tu_i=1$　③ $u_i^Tu_j=0$ for $i\neq j$

由此，$\exists$ parameters $(\beta_i)_{i=1}^n$，s.t. $x=\sum_{i=1}^n\beta_iu_i$，可以推出

① $A^TAx=A^TA\sum_{i=1}^n\beta_iu_i=\sum_{i=1}^n\lambda_i\beta_iu_i$

② $x^TA^TAx=(\sum_{i=1}^n\beta_iu_i)^T(\sum_{i=1}^n\lambda_i\beta_iu_i)=\sum_{i=1}^n\lambda_i\beta_i^2$

③ $\|x\|_2^2=x^Tx=(\sum_{i=1}^n\beta_iu_i)^T(\sum_{i=1}^n\beta_iu_i)=\sum_{i=1}^n\beta_i^2$ 有

$$\|A\|_2^2=\max_{\|x\|_2=1}\|Ax\|_2^2=\max_{\|x\|_2=1}x^TA^TAx=\max_{\|x\|_2=1}\sum_{i=1}^n\lambda_i\beta_i^2$$

$$\leq\lambda_{max}(A^TA)\sum_{i=1}^n\beta_i^2=\lambda_{max}(A^TA)\|x\|_2^2=\lambda_{max}(A^TA)$$

4. Frobenius norm $\|A\|_F=\left(\sum_{i=1}^m\sum_{j=1}^n|a_{ij}|^2\right)^{\frac12}=\sqrt{tr(A^TA)}$

① F-norm 满足 sub-multiplicative：$\|AB\|_F\leq\|A\|_F\|B\|_F$

② $\|\cdot\|_F$ is compatible with $\|\cdot\|_2$：$\|Ax\|_F\leq\|A\|_F\|x\|_2$

证 Denote $A=(a_{ij})_{i,j}$，$B=(b_{ij})_{i,j}$，则 $AB=(\sum_{k=1}^na_{ik}b_{kj})_{i,j}$

$$\|AB\|_F^2=\sum_{i,j}(\sum_{k=1}^na_{ik}b_{kj})^2\leq\sum_{i,j}(\sum_{k=1}^na_{ik}^2)\cdot(\sum_{k=1}^nb_{kj}^2)$$

$$=(\sum_{i,k}a_{ik}^2)(\sum_{k,j}b_{kj}^2)=\|A\|_F^2\cdot\|B\|_F^2$$

证 Trace 有以下性质：$trace(AB)=trace(BA)$

Eigenvalue decomposition 有 $A^TA=U\Sigma U^T$.

$\Sigma=diag(\lambda_1,\cdots,\lambda_n)$，$U=[u_1,u_2,\cdots,u_n]$，$U^TU=UU^T=I$

$$\|AB\|_F^2=trace(B^TA^TAB)=trace(BB^TA^TA)=trace(BB^TU\Sigma U^T)$$

$$=trace(U^TBB^TU\Sigma)\leq\lambda_{max}(A^TA)\,trace(U^TBB^TU)$$

$$=\lambda_{max}(A^TA)\,trace(UU^TBB^T)=\|A\|_2^2\,trace(BB^T)=\|A\|_2^2\|B\|_F^2$$

§3 Condition number

1. Condition number 一个 nonsingular square matrix A 的 condition number 为：

$$\text{cond}(A):=\|A\|\cdot\|A^{-1}\|$$

① 若 A 为 singular，则 cond(A) = ∞　② 使用 induced matrix norms

③ $\text{cond}(A)=\left(\max_{x\neq 0}\frac{\|Ax\|}{\|x\|}\right)\cdot\left(\min_{x\neq 0}\frac{\|Ax\|}{\|x\|}\right)^{-1}$

大 condition number ⇒ matrix nearly singular

证 $\|A^{-1}\|=\left(\max_{x\neq 0}\frac{\|A^{-1}x\|}{\|x\|}\right)=\left(\max_{y\neq 0}\frac{\|A^{-1}Ay\|}{\|Ay\|}\right)$

$$=\left(\max_{y\neq 0}\frac{\|y\|}{\|Ay\|}\right)=\left[\left(\max_{y\neq 0}\frac{\|y\|}{\|Ay\|}\right)^{-1}\right]^{-1}=\left(\min_{y\neq 0}\frac{\|Ay\|}{\|y\|}\right)^{-1}$$

2. Condition number 的性质

① $\text{cond}(A)=\|A\|\cdot\|A^{-1}\|\geq\|AA^{-1}\|=\|I\|=1$，$\forall$ A

② $\text{cond}(I)=1$　③ $\text{cond}(\alpha A)=\text{cond}(A)$ $\forall$ matrix A 与 $\alpha\in R$

④ $\forall$ diagonal matrix $D=diag(d)$，$\text{cond}(D)=\frac{\max_{i\in\{1,\cdots,n\}}|d_i|}{\min_{i\in\{1,\cdots,n\}}|d_i|}$

3. 计算对称矩阵的 $\text{cond}_{\|\cdot\|_2}$ 若 A symmetric 有特征值 $\{\lambda_1,\cdots,\lambda_n\}$，则 $\text{cond}_{\|\cdot\|_2}(A)=\frac{\max\{|\lambda_1|,\cdots,|\lambda_n|\}}{\min\{|\lambda_1|,\cdots,|\lambda_n|\}}$

§4 Assessing accuracy

1. Error bound (关于 b) ($Ax=b$，$A\hat{x}=b+\Delta b$)

$$\frac{\|\Delta x\|}{\|x\|}\leq\text{cond}(A)\cdot\frac{\|\Delta b\|}{\|b\|}$$

证 $\frac{\|\Delta x\|}{\|x\|}=\frac{\|\hat{x}-x\|}{\|x\|}=\frac{\|A^{-1}(b+\Delta b)-A^{-1}b\|}{\|A^{-1}b\|}=\frac{\|A^{-1}\Delta b\|}{\|A^{-1}b\|}$

$$\leq\|A^{-1}\|\cdot\frac{\|\Delta b\|}{\|A^{-1}b\|}=\|A^{-1}\|\cdot\frac{\|b\|}{\|A^{-1}b\|}\cdot\frac{\|\Delta b\|}{\|b\|}$$

$$\leq\|A^{-1}\|\cdot\|A\|\cdot\frac{\|\Delta b\|}{\|b\|}=\text{cond}(A)\cdot\frac{\|\Delta b\|}{\|b\|}$$

2. Error bound (关于 A) ($Ax=b$，$(A+\Delta A)\hat{x}=b$)

$$\frac{\|\Delta x\|}{\|\hat{x}\|}\leq\text{cond}(A)\cdot\frac{\|\Delta A\|}{\|A\|}$$

证 $\frac{\|\Delta x\|}{\|\hat{x}\|}=\frac{\|\hat{x}-x\|}{\|\hat{x}\|}=\frac{\|A^{-1}(b-\Delta A\hat{x})-A^{-1}b\|}{\|\hat{x}\|}=\frac{\|A^{-1}\Delta A\hat{x}\|}{\|\hat{x}\|}$

$$\leq\|A^{-1}\|\cdot\|\Delta A\|\cdot\frac{\|\hat{x}\|}{\|\hat{x}\|}=\|A^{-1}\|\cdot\|A\|\cdot\frac{\|\Delta A\|}{\|A\|}=\text{cond}(A)\cdot\frac{\|\Delta A\|}{\|A\|}$$

3. Conditioning 与 condition number

对于 problem $f(b)=A^{-1}b\ (=x)$，定义 condition number 为

$$\text{cond}=\frac{|\text{rel. forward error}|}{|\text{rel. backward error}|}=\frac{\|f(\hat{b})-f(b)\|/\|f(b)\|}{\|\hat{b}-b\|/\|b\|}=\frac{\|\Delta x\|/\|x\|}{\|\Delta b\|/\|b\|}$$

由此可知，cond(A) 为 cond 的 "uniform" worst-case bound

$$\sup_{b,\Delta b}\text{cond}=\sup_{b,\Delta b}\frac{\|\Delta x\|/\|x\|}{\|\Delta b\|/\|b\|}=\text{cond}(A)$$

因此，cond(A) 为 independent of b, $\hat{b}$ 的一个 version of cond

4. Some useful tools

- The representation of $AB=(\sum_{k=1}^na_{ik}b_{kj})_{i,j}$
- Cauchy-Schwartz inequality $(\sum_{k=1}^na_{ik}b_{kj})^2\leq(\sum_{k=1}^na_{ik}^2)(\sum_{k=1}^nb_{kj}^2)$
- Eigenvalue-eigenvector decomposition of real symmetric matrixes $A=U\Sigma U^\top$
- Property of trace $trace(AB)=trace(BA)$

4. Solving Linear Systems

§1 Triangular linear system

1. triangular matrix ① Lower: $a_{ij}=0$ for $i<j$ ② Upper: $a_{ij}=0$ for $i>j$

2. Forward substitution

① 对于 lower triangular system $L\mathbf{x}=\mathbf{b}$, forward substitution 为:

$$x_1 = b_1/l_{11};\ x_i = \left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j\right)/l_{ii},\ i=2,\cdots,n$$

②

```
for j = 1 to n                 /* loop over columns */
    if ℓ_jj = 0 then stop      /* stop if matrix is singular */
    x_j = b_j/ℓ_jj             /* compute solution component */
    for i = j+1 to n
        b_i = b_i − ℓ_ij x_j   /* update right-hand side */
    end
end
```

1 flop ← $x_j = b_j/\ell_{jj}$; $n-j$ flops; (overwrite the vector $\mathbf{b}$)

↳ 在计算出第j行的未知数 x_j 后，对于第j行以下的第i行，用 b_i 减去 $x_j \cdot x_j$ 在第i行的系数 l_{ij}

③ Total flops $\sum_{j=1}^{n} 2(n-j)+1 = n^2$

3. Backward substitution

① 对于 upper triangular system $U\mathbf{x}=\mathbf{b}$, backward substitution 为:

$$x_n = b_n/u_{nn};\quad x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij}x_j\right)/u_{ii},\ i=n-1,\cdots,1$$

②

```
for j = n to 1                 /* loop backwards over columns */
    if u_jj = 0 then stop      /* stop if matrix is singular */
    x_j = b_j/u_jj             /* compute solution component */
    for i = 1 to j−1
        b_i = b_i − u_ij x_j   /* update right-hand side */
    end
end
```

(overwrite the vector $\mathbf{b}$)

↳ 在计算出第j行的未知数 x_j 后，对于第j行以上的第i行，用 b_i 减去 $x_j \cdot x_j$ 在第i行的系数 u_{ij}

③ Total flops $\sum_{j=1}^{n} 2(n-j)+1 = n^2$

§2 Transform general systems

1. Permutation: $P^{-1}=P^T$

① $PA\mathbf{x}=P\mathbf{b}$ 相当于对 A 和 $\mathbf{b}$ 的 rows 进行 reorder, solution $\mathbf{x}$ 不变

② $AP\mathbf{x}=\mathbf{b}$ 相当于对 A 的 columns 进行 reorder, solution $\mathbf{x}=P^T(A^{-1}\mathbf{b})$ 改变

2. diagonal scaling (nonsingular diagonal matrix D)

① $DA\mathbf{x}=D\mathbf{b}$ 相当于对 A 的 rows 和 $\mathbf{b}$ 的乘以 D 的 diagonal entry, $\mathbf{x}$ 不变

② $AD\mathbf{x}=\mathbf{b}$ 相当于对 A 的 columns 乘以 D 的 diagonal entry, $\mathbf{x}=D^{-1}(A^{-1}\mathbf{b})$

§3 Elementary elimination matrix

1.
$$M_k\mathbf{a} = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{k+1} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_n & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$m_i = \frac{a_i}{a_k}$ for $\forall i = k+1,\cdots,n$

a_k 被称为 pivot, 必须为 nonzero

M_k 为 elementary elimination matrix

2. Elementary elimination matrix 的性质

① $M_k = I - \mathbf{m}_k\mathbf{e}_k^T$, $\mathbf{m}_k = [0,\cdots,m_{k+1},\cdots,m_n]^T$, $\mathbf{e}_k$ 为 k-th unit vector

② $L_k := M_k^{-1} = I + \mathbf{m}_k\mathbf{e}_k^T$

Sherman-Morrison Formula: $(A+\mathbf{b}\mathbf{c}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{b}\mathbf{c}^TA^{-1}}{1+\mathbf{c}^TA^{-1}\mathbf{b}}$

③ $M_kM_j = I - \mathbf{m}_k\mathbf{e}_k^T - \mathbf{m}_j\mathbf{e}_j^T\ (k<j)$ (M_k 与 M_j 的 "union")

④ $L_kL_j = I + \mathbf{m}_k\mathbf{e}_k^T + \mathbf{m}_j\mathbf{e}_j^T\ (k<j)$ (L_k 与 L_j 的 "union")

可推广至 $L_1L_2\cdots L_n = I + \sum_{i=1}^{n}\mathbf{m}_i\mathbf{e}_i^T$

§4 LU Factorization by Gaussian elimination

1. Gaussian elimination

$M_{n-1}\cdots M_1A\mathbf{x} = M_{n-1}\cdots M_1\mathbf{b} \Rightarrow MA\mathbf{x}=M\mathbf{b}$ 其中 $MA=U$

用 back-substitution 得到最终结果

2. LU Factorization

① 矩阵 $L_i = M_i^{-1}$ 为 unit lower triangular

② $M = M_{n-1}\cdots M_1 \Rightarrow L = M^{-1} = M_1^{-1}\cdots M_{n-1}^{-1} = L_1\cdots L_{n-1}$

注: 根据 L_i 的性质, $L = L_1\cdots L_{n-1}$ 可视作 $L_1,\cdots,L_{n-1}$ 的 union

③ 由 Gaussian elimination $MA=U \Rightarrow A = LU$, L 为 unit Δ

3. 利用 LU factorization 解 linear system

$LU\mathbf{x}=\mathbf{b} \Rightarrow$ solve $L\mathbf{y}=\mathbf{b}$ (forward sub) $\Rightarrow$ solve $U\mathbf{x}=\mathbf{y}$ (backward sub)

4. LU factorization algorithm

1° 逐步分析

① Step 1: $A^1 = \underbrace{\begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{m}_1 & I \end{bmatrix}}_{M_1} \underbrace{\begin{bmatrix} a_{11} & A(1,2:n) \\ * & A(2:n,2:n) \end{bmatrix}}_{A} = \begin{bmatrix} a_{11} & A(1,2:n) \\ \mathbf{0} & \mathbf{m}_1A(1,2:n)+A(2:n,2:n) \end{bmatrix}$

其中 $\mathbf{m}_1 = -A(2:n,1)/a_{11}$

② Step 2: $A^2 = \begin{bmatrix} 1 & 0 & \mathbf{0}^T \\ 0 & 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{m}_2 & I \end{bmatrix}\begin{bmatrix} a_{11}^1 & a_{12}^1 & A^1(1,3:n) \\ 0 & a_{22}^1 & A^1(2,3:n) \\ \mathbf{0} & * & A^1(3:n,3:n) \end{bmatrix} = \begin{bmatrix} a_{11}^1 & a_{12}^1 & A^1(1,3:n) \\ 0 & a_{22}^1 & A^1(2,3:n) \\ \mathbf{0} & \mathbf{0} & \mathbf{m}_2A^1(2,3:n)+A^1(3:n,3:n) \end{bmatrix}$

其中 $\mathbf{m}_2 = -A^1(3:n,2)/a_{22}^1$

- overwrite A, 使 output 矩阵的 upper triangular part 对应 U
- 将 $\mathbf{l}_k = -\mathbf{m}_k$ 储存在 output 矩阵的 lower triangular part

2° Algorithmic procedure

```
for k = 1 to n−1                 /* loop over columns */
    if a_kk = 0 then stop        /* stop if pivot is zero */
    for i = k+1 to n             /* compute multipliers
        ℓ_ik = a_ik/a_kk            for current column */
    end
    for j = k+1 to n
        for i = k+1 to n         /* apply transformations
            a_ij = a_ij − ℓ_ik a_kj    to remaining submatrix */
        end
    end
end
```

$\ell_{ik} = a_{ik}/a_{kk} = -\mathbf{m}_k$

Total flops: $\frac{2}{3}n^3 + O(n^2) = O(n^3)$

证 每轮 iter: $n-(k+1)+1$ (解 $\mathbf{l}_k = -\mathbf{m}_k$) $+ 2(n-k)^2$ (更新 block) $= (n-k)+2(n-k)^2$

共需要 $\sum_{k=1}^{n-1}(n-k)+2(n-k)^2 = \sum_{i=1}^{n-1}2i^2+i = 2\cdot\frac{n(n+1)(2n+1)}{6} + \frac{n(n-1)}{2} = \frac{2}{3}n^3+O(n^2)$

5. 比较: inversion 与 factorization

① 计算 A^{-1} 的方法: 求解 $A^{-1} \iff$ 求解 $A\mathbf{x}_i = \mathbf{e}_i$ for all i

② 计算 A^{-1} 的 complexity: (对 A LU分解 + n次 forward & backward substitution) $O(n^3)$

③ 比较: inversion 与 factorization

(1) Initial cost: LU factorization of A < 计算 A^{-1}

(2) Later cost: forward & backward substitution ≈ 计算 $A^{-1}b_i = O(n^2)$

§5 Partial pivoting

1. Pivoting strategy: partial pivoting

每个 stage, 选取该 column 对角线以下绝对值最大的 entry 作为 pivot

(↓ error propagation) 每个 M_k 前都要乘以 permutation matrix P_k 以交换两行

2. Partial pivoting: 第一种方法 仍然得到 $MA=U$, 此时

① $M = M_{n-1}P_{n-1}\cdots M_1P_1$ ② $L = M^{-1}$ 不一定为 lower triangular

3. Partial pivoting: 第二种方法 希望得到 $PA=LU$, 此时

① $P = P_{n-1}\cdots P_1$ 对 A 的 rows 重排列 ② $L = PM^{-1}$ 为 lower triangular

1° 逐步分析 (step k) ($A' = P_k\cdots P_1A$)

$P_kA_{k-1}' = P_kL_k'P_k^TP_kU_k'$ (L' 合并了之前所有的 L_i, P_i, P_i^T)

由于 $L_k = I + \mathbf{m}_k\mathbf{e}_k^T$, 对于 $\forall j>k$, 有

$P_jL_kP_j^T = P_jP_j^T + P_j\mathbf{m}_k\mathbf{e}_k^TP_j^T = I + (P_j\mathbf{m}_k)\mathbf{e}_k^T = \tilde{L}_k$

$P_kL_{k-1}'P_k^T = P_k(\tilde{L}_{k-1}\tilde{L}_{k-2}\cdots\tilde{L}_1)P_k^T = (P_k\tilde{L}_{k-1}P_k^T)(P_k\tilde{L}_{k-2}P_k^T)\cdots(P_k\tilde{L}_1P_k^T)$

相当于交换了 L_i 对应的两行 multiplier

eg. 对下列矩阵做 LU factorization (with pivoting) $A = \begin{bmatrix} 1 & -4 & 2 \\ -2 & 3 & 0 \\ 4 & -1 & 2 \end{bmatrix}$

① swap row 1 & 3 + 求解 L_1 & U_1

$P_1A = \begin{bmatrix} 4 & -1 & 2 \\ -2 & 3 & 0 \\ 1 & -4 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1/2 & 1 & 0 \\ 1/4 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 & -1 & 2 \\ 0 & 5/2 & 1 \\ 0 & -15/4 & 3/2 \end{bmatrix}$

② swap row 2 & 3 + 求解 L_2 & U_2

$P_2P_1A = \begin{bmatrix} 4 & -1 & 2 \\ 1 & -4 & 2 \\ -2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 & -1 & 2 \\ 0 & -15/4 & 3/2 \\ 0 & 5/2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ -1/2 & -2/3 & 1 \end{bmatrix}\begin{bmatrix} 4 & -1 & 2 \\ 0 & -15/4 & 3/2 \\ 0 & 0 & 2 \end{bmatrix}$

③ 因此, $PA = \begin{bmatrix} 4 & -1 & 2 \\ 1 & -4 & 2 \\ -2 & 3 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ -1/2 & -2/3 & 1 \end{bmatrix}\begin{bmatrix} 4 & -1 & 2 \\ 0 & -15/4 & 3/2 \\ 0 & 0 & 2 \end{bmatrix}$

2° Total flops: $\frac{2}{3}n^3 + O(n^2)$

§6 LU Factorization: Summary & Stability

1. LU Factorization with pivoting 的存在性定理

若 $A\in\mathbb{R}^{n\times n}$ nonsingular matrix, 则 $\exists\, PA=LU$ 且 $|l_{ij}|\le 1, \forall i,j$ & L 为 unit Δ

注: 该定理为存在性定理, LU分解可能不唯一.

证 ① Base case: $n=1$ 对于 $A\in\mathbb{R}$, 有 $A = 1\cdot A$, 因此选取 $P=L=1$, $U=A$ 即可

② Inductive step: $n-1\to n$

令 $A\in\mathbb{R}^{n\times n}$ 为 nonsingular, 则其第一列 $A(:,1)$ 不全为0, 因此,

$\alpha := \max_{i=1,\cdots,n}|A(i,1)| \ne 0$, $\exists\, P_1A = \begin{bmatrix} \alpha & \mathbf{u}^T \\ \mathbf{v} & A_* \end{bmatrix}$

进行 LU-factorization 的第一步, 有

$P_1A = \begin{bmatrix} \alpha & \mathbf{u}^T \\ \mathbf{v} & A_* \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \\ \mathbf{v}/\alpha & I \end{bmatrix}}_{=L_1}\begin{bmatrix} \alpha & \mathbf{u}^T \\ \mathbf{0} & A_*-\mathbf{v}\mathbf{u}^T/\alpha \end{bmatrix}$

$B = A_* - \mathbf{v}\mathbf{u}^T/\alpha$ 也为 nonsingular, $P_*B = P_*(A_* - \frac{1}{\alpha}\mathbf{v}\mathbf{u}^T) = L_*U_*$

$\underbrace{\begin{bmatrix} 1 & \\ & P_* \end{bmatrix}P_1}_{P}A = \begin{bmatrix} 1 & \\ & P_* \end{bmatrix}\begin{bmatrix} 1 & \\ \mathbf{v}/\alpha & I \end{bmatrix}\begin{bmatrix} 1 & \\ & P_*^T \end{bmatrix}\begin{bmatrix} 1 & \\ & P_* \end{bmatrix}\begin{bmatrix} \alpha & \mathbf{u}^T \\ \mathbf{0} & B \end{bmatrix}$

$= \underbrace{\begin{bmatrix} 1 & \\ P_*\mathbf{v}/\alpha & L_* \end{bmatrix}}_{L}\underbrace{\begin{bmatrix} \alpha & \mathbf{u}^T \\ \mathbf{0} & U_* \end{bmatrix}}_{U}$

2. Pivoting & stability

① Gaussian elimination without pivoting 为 unstable; with pivoting 为 stable

② accuracy 仍取决于 A 的 conditioning

§7 Cholesky factorization

1. Cholesky factorization 若 $A=A^T$ and $\mathbf{x}^TA\mathbf{x}>0$, $\forall\mathbf{x}\ne 0$ 则 $A=LL^T$

2. Cholesky factorization 的计算

① 2×2 case: $l_{11}=\sqrt{a_{11}}$, $l_{21}=a_{21}/l_{11}$, $l_{22}=\sqrt{a_{22}-l_{21}^2}$

② n×n case: $\hat{A} = \begin{bmatrix} LL^T & L\mathbf{l} \\ \mathbf{l}^TL^T & l_{nn}^2+\|\mathbf{l}\|^2 \end{bmatrix}$, $\mathbf{l} = L^{-1}\mathbf{a}$ 与 $l_{nn} = \sqrt{a_{nn}-\|\mathbf{l}\|^2}$

3. Cholesky algorithm 的 features ① 不需要 pivoting ② 仅需储存 upper(或lower) triangular part ③ Total flops: $\frac{1}{3}n^3+O(n^2)$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

5 Linear Least Square

§1 Linear least square

1. Linear least square 对于 overdetermined linear system: $A\mathbf{x}=\mathbf{b}$, $A\in\mathbb{R}^{m\times n}$, $m>n$

Least squares 求解 $\mathbf{x}$ 来 minimize residual vector $\mathbf{r} = A\mathbf{x}-\mathbf{b}$ 的 $\|\cdot\|_2$:

$\min_x \|\mathbf{r}\|_2^2 = \min_x \|A\mathbf{x}-\mathbf{b}\|_2^2$

2. Euclidean projection of $\mathbf{b}$ onto range(A)

$P_{range(A)}(\mathbf{b}) = \mathbf{y}^* := \arg\min_{\mathbf{y}\in range(A)} \frac{1}{2}\|\mathbf{y}-\mathbf{b}\|_2^2$

由于 $\mathbf{y}^*\in range(A)$, 则 $\exists\,\mathbf{x}^*$ s.t. $\mathbf{y}^* = A\mathbf{x}^*$

证 uniqueness: 利用 p 为 quadratic, second-order Taylor expansion 为 exact

$p(\mathbf{y}) = p(\mathbf{y}^*) + \underbrace{(\mathbf{y}^*-\mathbf{b})^T(\mathbf{y}-\mathbf{y}^*)}_{=0\ \text{(optimality condition)}} + \frac{1}{2}(\mathbf{y}-\mathbf{y}^*)^T(\mathbf{y}-\mathbf{y}^*) \Rightarrow \|\mathbf{y}-\mathbf{y}^*\| = 0$

证 existence: 先将 $\mathbf{y}^*$ 的范围缩到 compact set 上, 再利用极值定理

$L := \{\mathbf{y}: p(\mathbf{y}) \le p(\mathbf{0})\} \cap range(A)$

3. least square solution 对于 least square problem $\min_x \|A\mathbf{x}-\mathbf{b}\|_2^2$

① 一定有 solution ② $\mathbf{x}^*$ 为 optimal iff 满足 normal equations: $A^TA\mathbf{x}^* = A^T\mathbf{b}$

③ solution 为 unique 当且仅当 A 满秩 (rank(A) = n) $\mathbf{x}^* = (A^TA)^{-1}A^T\mathbf{b}$

若 rank < n, 即 A 为 rank-deficient, 则 solution 不唯一

证 "⇒" 令 $f(\mathbf{x}) = \|\mathbf{r}\|_2^2 = (A\mathbf{x}-\mathbf{b})^T(A\mathbf{x}-\mathbf{b}) = \mathbf{x}^TA^TA\mathbf{x} - 2\mathbf{x}^TA^T\mathbf{b} + \mathbf{b}^T\mathbf{b}$

若 $\mathbf{x}^*$ 为 (local) solution, 其必须满足 $\nabla f(\mathbf{x}^*) = 2A^TA\mathbf{x}^* - 2A^T\mathbf{b} = 0$

"⇐" 令 $\mathbf{x}^*$ 为 normal equation 的一个 solution, 即 $f'(\mathbf{x}) = 0$

$f(\mathbf{x}) - f(\mathbf{x}^*) = \nabla f(\mathbf{x}^*)^T(\mathbf{x}-\mathbf{x}^*) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^T\nabla^2 f(\mathbf{x})(\mathbf{x}-\mathbf{x}^*)$

$= 0 + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^T\cdot 2A^TA(\mathbf{x}-\mathbf{x}^*) = \|A(\mathbf{x}-\mathbf{x}^*)\|_2^2 \ge 0, \forall\mathbf{x}\in\mathbb{R}^n$

§2 Orthogonal projectors & conditioning

1. Orthogonal projectors ① $P^2=P$ ② $P^T=P$

2. vector 的分解 $P_\perp = I-P$ 为 orthogonal projector onto range$(P)^\perp$

3. 最小二乘问题的正交投影算子 $P_{range(A)} = P = A(A^TA)^{-1}A^T$

① $\mathbf{b} = P\mathbf{b} + P_\perp\mathbf{b} = A\mathbf{x}^* + (\mathbf{b}-A\mathbf{x}^*) = \mathbf{y}^* + (-\mathbf{r})$

② $\mathbf{y}^* = P_{range(A)}(\mathbf{b}) = A\mathbf{x}^* = A(A^TA)^{-1}A^T\mathbf{b} = P_{range(A)}\cdot\mathbf{b}$

4. Pseudoinverse 对于 non-square m×n matrix A

若 rank(A) = n 则 pseudoinverse 为: $A^+ = (A^TA)^{-1}A^T$

注: $\mathbf{x}^* = (A^TA)^{-1}A^T\mathbf{b} = A^+\mathbf{b}$

5. condition number 对于 non-square m×n matrix A

若 rank(A) = n 则 condition number 为: cond$(A) := \|A\|_2\cdot\|A^+\|_2$

① 若 rank(A) < n, 则 cond$(A) := \infty$

② condition number 衡量了 closeness to rank deficiency

6. Conditioning 与 sensitivity (条件数的应用) θ 为 $\mathbf{b}$ 与 $\mathbf{y} = A\mathbf{x}$ 的夹角

$\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} \le \frac{cond(A)}{\cos(\theta)}\cdot\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2}$, $\cos(\theta) = \frac{\|\mathbf{y}\|_2}{\|\mathbf{b}\|_2}$

6. Solving Least Square

§1 Normal equation method

1. Normal equation method: 用 $A^TA = LL^T$ 解 $A^TA\mathbf{x} = A^T\mathbf{b}$

2. 缺点: ① 求 A^TA 与 $A^T\mathbf{b}$ 时 lose information ② cond$(A^TA) = (\text{cond}(A))^2$

§2 Orthogonalization methods

1. Orthogonal matrix ① Q 的列向量为 $\mathbb{R}^n$ 的规范正交基 ② $Q^{-1}=Q^T$

③ $\|Q\mathbf{x}\| = \|\mathbf{x}\|$ ④ $\langle Q\mathbf{x}, Q\mathbf{y}\rangle = \langle\mathbf{x},\mathbf{y}\rangle$ ⑤ AB 也为正交矩阵

2. Orthogonal transformation $\min_x\|A\mathbf{x}-\mathbf{b}\|_2^2 = \min_x\|TA\mathbf{x}-T\mathbf{b}\|_2^2$

$T_{m\times m}A_{m\times n} = \begin{bmatrix} R_{n\times n} \\ 0_{(m-n)\times n} \end{bmatrix}$, $T_{m\times m}\mathbf{b}_{m\times 1} = \begin{bmatrix} \mathbf{c}_{1,n\times 1} \\ \mathbf{c}_{2,(m-n)\times 1} \end{bmatrix}$ $= \min_x\|R\mathbf{x}-\mathbf{c}_1\|_2^2 + \|\mathbf{c}_2\|_2^2 \iff$ 解 $R\mathbf{x} = \mathbf{c}_1$ $= \|A\mathbf{x}-\mathbf{b}\|_2^2 = \|\mathbf{r}\|_2^2$

3. QR Factorization $A_{m\times n} = Q_{m\times m}\begin{bmatrix} R_{n\times n} \\ 0_{(m-n)\times n} \end{bmatrix}$ $\min_x\|A\mathbf{x}-\mathbf{b}\|_2^2 \iff \min_x\|R\mathbf{x}-\mathbf{c}_1\|_2^2$ 其中 $Q^T\mathbf{b} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}$

4. Reduced QR Factorization $Q_{m\times m} = [Q_{r\ m\times n}\ \ Q_{n\ m\times(m-n)}]$

$A = Q\begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_r\ Q_n]\begin{bmatrix} R \\ 0 \end{bmatrix} = Q_rR$ $\min_x\|A\mathbf{x}-\mathbf{b}\|_2^2 \iff R\mathbf{x} = \mathbf{c}_1\ (= Q_r^T\mathbf{b})$

5. Reduced QR Factorization 的性质 ① A 有 full rank $\iff r_{ii}\ne 0, \forall i$

② 若 A 有 full rank, Q_r 的 columns 为 range(A) 的 orthonormal basis

Q_n 的 columns 为 range$(A)^\perp$ 的 orthonormal basis

③ $Q_rQ_r^T$ 为 range(A) 的 orthogonal projector

§3 Householder QR factorization

1. reflection matrix: $H = I - 2P$ { ① $H = H^T = H^{-1}$ ② $H\mathbf{x}$ 将 $\mathbf{x}$ 关于 range$(P_\perp)$ reflect }

2. Householder matrix: $H_v = I - 2\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|_2^2}$ for $\mathbf{v}\in\mathbb{R}^m\setminus\{0\}$

① $\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2}$ 为 orthogonal projector ② $H_v = H_v^T = H_v^{-1}$ ③ $\begin{bmatrix} I_r & \mathbf{0}^T \\ \mathbf{0} & H_v \end{bmatrix}$

3. Householder transformation

① $\alpha = \pm\|\mathbf{a}\|_2$ ② $\mathbf{v} = \mathbf{a} - \alpha\mathbf{e}_1 \iff H_v\mathbf{a} = \alpha\mathbf{e}_1$

4. Householder QR factorization (stable)

```
for k = 1 to n
    Set A = Q_k A, where                 /* update A */
        Q_k = [ I_{k-1}        ]
              [         H_{v_k} ]
    I_{k-1} is (k−1)-dim. identity matrix; H_{v_k} is the Householder trans-
    formation using the first column of A_{k−1} to build v_k.
end
```

实际的 update ① 求 $\mathbf{w} = \frac{\sqrt{2}\mathbf{v}_k}{\|\mathbf{v}_k\|}$ ② $H_{v_k}A_{k-1} = A_{k-1} - \mathbf{w}(\mathbf{w}^TA_{k-1}) = \begin{bmatrix} \alpha_k & * \\ \mathbf{0} & A_k \end{bmatrix}$

① $Q = (Q_n\cdots Q_1)^T = Q_1^T\cdots Q_n^T = Q_1\cdots Q_n$

② Flops $= \sum_{k=1}^{n} 3(m-k+1)+1+[4(m-k+1)-1](n-k) \approx 2mn^2 - \frac{2}{3}n^3$

③ normal equation method: $mn^2 + \frac{1}{3}n^3$ flops

§4 Givens QR Factorization

$$G_{i,k}\mathbf{a} = \begin{bmatrix} I & & & & \\ & c^i & & s^k & \\ & & I & & \\ & -s^k & & c^i & \\ & & & & I \end{bmatrix}\mathbf{a} = \begin{bmatrix} \\ \alpha \\ \\ 0 \\ \\ \end{bmatrix} \begin{matrix} \\ \leftarrow i \\ \\ \leftarrow k \\ \\ \end{matrix}$$

$s = \frac{a_k}{\sqrt{a_i^2+a_k^2}}$, $c = \frac{a_i}{\sqrt{a_i^2+a_k^2}}$, $\alpha = \sqrt{a_i^2+a_k^2}$

$\underbrace{(G_{n,m}\cdots G_{n,n+2}G_{n,n+1})\cdots(G_{2,m}\cdots G_{2,4}G_{2,3})(G_{1,m}\cdots G_{1,3}G_{1,2})}_{=:Q^T}A = R$

Use Givens transformation to do the QR factorization of matrix

$A = \begin{pmatrix} 4 & 2 \\ 3 & 4 \\ 0 & \sqrt{5} \end{pmatrix}$

Step 1: Find c_{12}, s_{12} s.t. $G(1,2,c_{1,2},s_{1,2})A_{:,1} = (*\ 0\ *)^T$

a) $c_{1,2} = \frac{A_{1,1}}{\sqrt{A_{1,1}^2+A_{2,1}^2}} = \frac{4}{5}$, $s_{1,2} = \frac{A_{2,1}}{\sqrt{A_{1,1}^2+A_{2,1}^2}} = \frac{3}{5}$

b) $\tilde{A} = G(1,2,c_{1,2},s_{1,2})A = \begin{pmatrix} c_{1,2}A_{1,:}+s_{1,2}A_{2,:} \\ -s_{1,2}A_{1,:}+c_{1,2}A_{2,:} \\ A_{3,:} \end{pmatrix} = \begin{pmatrix} 5 & 4 \\ 0 & 2 \\ 0 & \sqrt{5} \end{pmatrix}$

Step 2: Find $c_{1,3}, s_{1,3}$ s.t. $G(1,3,c_{1,3},s_{1,3})\tilde{A}_{:,1} = (*\ 0\ 0)^T$

$c_{1,3} = 1$, $s_{1,3} = 0$, $\tilde{A} = G(1,3,c_{1,3},s_{1,3})\tilde{A} = I\tilde{A} = \tilde{A}$

Step 3: Find c_{23}, s_{23} s.t. $G(2,3,c_{2,3},s_{2,3})\tilde{A}_{:,2} = (*\ *\ 0)^T$

$c_{2,3} = \frac{2}{3}$, $s_{2,3} = \frac{\sqrt{5}}{3}$, $R = G(2,3,c_{2,3},s_{2,3})\tilde{A} = \begin{pmatrix} 5 & 4 \\ 0 & 3 \\ 0 & 0 \end{pmatrix}$

G不满足 $G = G^T$

$Q = G_{1,2}^TG_{1,3}^TG_{2,3}^T$, $R = \begin{pmatrix} 5 & 4 \\ 0 & 3 \\ 0 & 0 \end{pmatrix}$

§5 Gram-Schmidt QR Factorization

1. Gram-Schmidt process 有线性无关组 $\{\mathbf{u}_1,\cdots,\mathbf{u}_m\}$, 求规范正交集 $\{\mathbf{v}_1,\cdots,\mathbf{v}_m\}$

① $\mathbf{v}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$ ② $\mathbf{p}_1 = \langle\mathbf{u}_2,\mathbf{v}_1\rangle\mathbf{v}_1$, $\mathbf{r}_1 = \mathbf{u}_2 - \mathbf{p}_1 \perp \text{Span}(\mathbf{u}_1)$. $\mathbf{v}_2 = \frac{\mathbf{r}_1}{\|\mathbf{r}_1\|} = \frac{\mathbf{u}_2-\mathbf{p}_1}{\|\mathbf{u}_2-\mathbf{p}_1\|}$

③ $\mathbf{p}_{m-1} = \langle\mathbf{u}_m,\mathbf{v}_1\rangle\mathbf{v}_1 + \cdots + \langle\mathbf{u}_m,\mathbf{v}_{m-1}\rangle\mathbf{v}_{m-1}$, $\mathbf{r}_{m-1} = \mathbf{u}_m - \mathbf{p}_{m-1} \perp \text{Span}(\mathbf{v}_1,\cdots,\mathbf{v}_{m-1})$

$\mathbf{v}_m = \frac{\mathbf{r}_{m-1}}{\|\mathbf{r}_{m-1}\|} = \frac{\mathbf{u}_m-\mathbf{p}_{m-1}}{\|\mathbf{u}_m-\mathbf{p}_{m-1}\|}$ Span$(\{\mathbf{u}_1,\cdots,\mathbf{u}_m\})$ = Span$(\{\mathbf{v}_1,\cdots,\mathbf{v}_m\})$

2. R 的求解 $r_{ij} := \mathbf{q}_i^T\mathbf{a}_j$ ($\mathbf{a}_j$ 在 $\mathbf{q}_i$ 上的投影长度), $i<j$

$r_{jj} := \|\tilde{\mathbf{q}}_j\| = \|\mathbf{a}_j - \sum_{i=1}^{j-1}r_{ij}\mathbf{q}_i\|$ ($\mathbf{a}_j$ 在 $\mathbf{q}_j$ 上的投影长度)

```
input: A collection of vectors a_1, ..., a_n
q̃_1 = a_1, q_1 = q̃_1/||q̃_1||.
for j = 2 to n
    q̃_j = a_j − Σ_{i=1}^{j−1} (q_i^T a_j) q_i.
    q_j = q̃_j/||q̃_j||.
end
output: q_1, ..., q_n.
```

$q_i^Ta_j = r_{ij}\ (i=1,\cdots,j-1)$; $\|\tilde{q}_j\| = r_{jj}$

$\mathbf{a}_j = [\mathbf{q}_1\ \cdots\ \mathbf{q}_j\ \cdots\ \mathbf{q}_n]_{m\times n}\begin{bmatrix} r_{1j} \\ \vdots \\ r_{jj} \\ \vdots \\ 0 \end{bmatrix}_{n\times 1}$ $A = Q_{m\times n}\begin{bmatrix} r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\ & \ddots & \vdots & & \vdots \\ & & r_{jj} & & r_{jn} \\ & & & \ddots & \vdots \\ 0 & & & & r_{nn} \end{bmatrix}_{n\times n}$

3. Classical Gram-Schmidt QR Factorization

```
for j = 1 to n
    q_j = a_j
    for i = 1 to j − 1
        r_ij = q_i^T a_j
        q_j = q_j − r_ij q_i
    end
    r_jj = ||q_j||_2
    q_j = q_j/r_jj
end
```

缺点 ① loss of orthogonality ② 需要分开储存 A, Q, 和 R

```
for i = 1 to n
    r_ii = ||a_i||_2
    q_i = a_i/r_ii
    for j = i + 1 to n
        r_ij = q_i^T a_j
        a_j = a_j − r_ij q_i
    end
end
```

通常更 stable

4. Modified Gram-Schmidt QR Factorization

$\tilde{\mathbf{q}}_j = (I - \mathbf{q}_{j-1}\mathbf{q}_{j-1}^T)\cdots(I-\mathbf{q}_2\mathbf{q}_2^T)(I-\mathbf{q}_1\mathbf{q}_1^T)\mathbf{a}_j = (I - \sum_{i=1}^{j-1}\mathbf{q}_i\mathbf{q}_i^T)\mathbf{a}_j$

在每次算出一个 $\mathbf{q}_j$ 后, 我们算出 ① $r_{j,j+1},\cdots,r_{j,n}$ ② $(I-\mathbf{q}_j\mathbf{q}_j^T)\mathbf{a}_j$

5. Flops: inner loop 中的 update: $(n-i)(4m-1)$; 总计: $\approx 2mn^2 - \frac{1}{2}n$

证 考虑 $\mathbf{y}^* = P_{range(A)}(\mathbf{b}) := \arg\min_{\mathbf{y}\in range(A)}\frac{1}{2}\|\mathbf{y}-\mathbf{b}\|_2^2$ $\exists\,\mathbf{x}^*$ s.t. $\mathbf{y}^* = A\mathbf{x}^*$

令 $p(\mathbf{y}) = \frac{1}{2}\|\mathbf{y}-\mathbf{b}\|_2^2$, 则 $\nabla p(\mathbf{y}^*)^T(\mathbf{y}-\mathbf{y}^*) = (\mathbf{y}^*-\mathbf{b})^T(\mathbf{y}-\mathbf{y}^*) = 0$,

$\forall\mathbf{y}\in range(A)$. 代入 $\mathbf{y}^* = A\mathbf{x}^*$, $\mathbf{y} = A\mathbf{x}$, 有

$(A\mathbf{x}^*-\mathbf{b})^TA(\mathbf{x}-\mathbf{x}^*) = (A^TA\mathbf{x}^*-A^T\mathbf{b})^T(\mathbf{x}-\mathbf{x}^*) = 0, \forall\mathbf{x}\in\mathbb{R}^n$

令 $\mathbf{x} = A^TA\mathbf{x}^* - A^T\mathbf{b} + \mathbf{x}^*$, 有 $\|A^TA\mathbf{x}^*-A^T\mathbf{b}\|^2 = 0$

§6 QR factorization 的存在性与唯一性

1. 存在性：所有 matrix $A\in R^{m\times n}(m\ge n)$ 均有 full QR factorization

注：① Gram-Schmidt 可得到 reduced QR factorization $A=\hat{Q}\hat{R}$，$\hat{Q}$ 可被扩展

② 若 A 为 rank deficient ($\text{rank}(A)<n$)，则 QR factorization 仍然存在

2. 唯一性：所有 matrix $A\in R^{m\times n}(m\ge n)$ will full rank 均有唯一的 reduced QR factorization $A=\hat{Q}\hat{R}$ with $r_{jj}>0$

注：① 通过 Gram-Schmidt 可以得到 unique 的 $\hat{Q}$ 与 R（除了 r_{jj} 的 sign）

§7 Rank deficiency

得到的R为 singular 最小二乘问题的解有多个 vector x

1. Pivoting：选取 remaining unreduced submatrix 中 $\|\cdot\|_2$ 最大的列构建 v_k

① 若 $\text{rank}(A)=r<n$，则 r 步以后，remaining unreduced 的 norms 会（接近）为 0

② Orthogonal factorization 的形式为

$$Q^TAP=\begin{bmatrix}R & S\\ 0 & 0\end{bmatrix}\ (R\in R^{r\times r})$$

R为一个 $r\times r$ 上三角 nonsingular 矩阵；P 为 column interchanges 的 permutation 矩阵

2. 解 basic solution：① 求解 $R_{r\times r}y=c_1$ 其中 $\begin{bmatrix}c_1\\ c_2\end{bmatrix}=Q^Tb$，$c_1\in R^r$

② 再求解 x：$x=P_{n\times n}\begin{bmatrix}y\\ 0\end{bmatrix}$

7. Eigenvalue problem

§1 Complex number

1. notations ① $\bar{\alpha}=a-bi$ ② $|\alpha|=\sqrt{a^2+b^2}=\sqrt{\alpha\bar{\alpha}}$

③ $\|z\|=\sqrt{z^Hz}$，$z^H=\bar{z}^T$ ④ $(AC)^H=C^HA^H$

2. conjugate transpose ① $(A^H)^H=A$ ② $(\alpha A+\beta B)^H=\bar{\alpha}A^H+\bar{\beta}B^H$

3. Hermitian matrix 若 $M^H=M$，则称 M 为 Hermitian matrix.

注：① Hermitian matrix 主对角线上的元素为实数.

② Hermitian matrix 的特征值为实数

③ Hermitian matrix 的积，逆矩阵也为 Hermitian matrix

4. unitary matrix U 的列向量构成 C^n 中的一个 orthonormal set

U is unitary $\iff U^HU=I \iff U^{-1}=U^H$

§2 Characteristic polynomial 与 multiplicity

1. Characteristic polynomial：$p(\lambda)=\det(A-\lambda I)=0$

注：① 一定有 n 个 eigenvalues，但eigenvalues不一定distinct 或 real

② 若 A real，且 eigenvalue λ real，则对应 eigenvector v 也 real

③ 一个 real matrix，若 $\alpha+\beta i$ 为 eigenvalue，则 $\alpha-\beta i$ 也是

2. Algebraic multiplicity & Geometric multiplicity

令 $\{\mu_1,\cdots,\mu_k\}$，$k\le n$ 表示 A 的 eigenvalue 的（distinct）取值的集合，则

① Algebraic multiplicity：$r_i=|\{\lambda_j:\lambda_j=\mu_i\}|$，$\forall i=1,\cdots,k$

② Geometric multiplicity：$\gamma_i=\dim(\text{null}(A-\mu_iI))$，$\forall i=1,\cdots,k$

注：subspace $\text{null}(A-\mu_iI)$ 被称为 μ_i eigenspace

③ 代数重数大于等于几何重数：$r_i\ge\gamma_i$ for all $i=1,\cdots,k$

3. defective matrix：若 $r_i>\gamma_i$ for some i

注：① $r_i=\dim(\text{null}[(A-\lambda_iI)^n])$ for $\forall$ large enough n

② 在 Jordan normal form 中，$r_i=\lambda_i$ 在 diagonal 中出现的次数

$\gamma_i=\lambda_i$ 对应的 Jordan block 数

§3 Eigendecomposition 与 diagonalizability

1. Eigendecomposition / diagonalizability：$\exists$ nonsingular $V\in C^{n\times n}$ s.t.

$$A=V\Lambda V^{-1}\Rightarrow AV=V\Lambda\Rightarrow Av_i=\lambda_iv_i$$

2. 一个性质：不同eigenvalue 的 eigenvectors 一定 linearly independent

证 令 $Av_1=\lambda_1v_1$，$Av_2=\lambda_2v_2$，且 $\lambda_1\ne\lambda_2$，$v_1,v_2\ne0$

假设 v_1 与 v_2 linearly dependent，则 $\exists\alpha\ne0$，s.t. $v_1=\alpha v_2$

$$\Rightarrow\begin{cases}Av_1=\lambda_1v_1=\lambda_1\alpha v_2\\ Av_1=\alpha Av_2=\alpha\cdot\lambda_2v_2\end{cases}\Rightarrow\lambda_1=\lambda_2\ (若\ \|v_2\|\ne0)$$ (contradiction)

3. Similarity transformation：$A,B\in C^{n\times n}$ similar 若 $\exists$ nonsingular $S\in C^{n\times n}$ s.t. $B=S^{-1}AS$，则

① A，B 的 eigenvalues 相同

② 若 u 为 B 的 eigenvector，则 $v=Su$ 为 A 的 eigenvector

③ A，B 的 characteristic equation，重数相同

注：A 有 eigendecomposition 若 A is similar to $\Lambda=\text{diag}(\lambda_1,\cdots,\lambda_n)$

证 ③：令 $B=S^{-1}AS$，则 $p_B(\lambda)=\det(S^{-1}AS-\lambda I)$
$=\det[S^{-1}(A-\lambda I)S]=\det(S^{-1})\det(A-\lambda I)\det(S)=\det(A-\lambda I)$

4. diagonalizability：$A=V\Lambda V^{-1}\iff$ A is not defective

证 "⇒" $A=V\Lambda V^{-1}\iff$ similar to Λ ∵ Λ non-def ∴ A non-def

证 "⇐" A non-def ⇒ 有 n 个 linearly independent 的 eigenvectors

∵ 与不同 eigenvalue 相关联的 eigenvectors 一定 linearly independent；每个 eigenvalue 可对应 $r_i=\gamma_i$ 个 linearly independent eigenvectors

∴ 因此可以排列 V 中的 eigenvectors 并写出 $A=V\Lambda V^{-1}$

5. Unitary diagonalization：$\exists$ unitary matrix $Q\in C^{n\times n}$ s.t. $A=Q\Lambda Q^H$

6. Unitary diagonalizability：Hermitian ($A^H=A$) unitarily diagonalizable 且所有 eigenvalues 均为 real.

§4 Schur factorization

1. Schur factorization：$A=QTQ^H$（A 与 T 的特征值相同）

2. 存在性：所有square matrix $A\in C^{n\times n}$ 均有 Schur factorization

§5 不改变 eigenvalue problem 的 transformations

1. Shifts：$(A-\sigma I)v=(\lambda-\sigma)v$（eigenvalues 也 shift σ）

2. Inversion：$A^{-1}v=\lambda^{-1}v$（eigenvalues 也 inverse）

3. Powers：$A^kv=\lambda^kv$（eigenvalues 也取 k-th power）

4. Polynomial transformation：$p(A)v=p(\lambda)v$

5. Similarity transformation：$A=Q\Lambda Q^H$，$A=QTQ^H$

§6 Power iteration / Power method

1. 基本介绍：重复对一个 nonzero initial vector x^0 左乘 matrix A

$x^k=Ax^{k-1}$. 若 A 有唯一的 eigenvalue of maximum modulus λ_1，其对应的 eigenvector 为 v_1. 则会收敛至 αv_1

2. 原理：① Assumptions：non-defective + $|\lambda_1|>|\lambda_2|\ge\cdots$

② Step 1：$x^0=\sum_{i=1}^n\alpha_iv_i$（A non-defective） ③ Step 2：

$$x^k=A^kx^0=\sum_{i=1}^n\lambda_i^k\alpha_iv_i=\lambda_1^k\Big[\alpha_1v_1+\sum_{i=2}^n\Big(\frac{\lambda_i}{\lambda_1}\Big)^k\alpha_iv_i\Big]$$

$|\cdot|<1\Rightarrow\to0$

3. Power iteration 的缺点：power iteration 会 fail：

① starting vector x^0 没有 v_1 方向的分量（即 $\alpha_1=0$）

② 不止一个 eigenvalue 有 maximum modulus. 会收敛至线性组合

③ 对于 real matrix A 和 real starting vector x^0，不会收敛至 complex vector

§7 Rayleigh quotient 与 normalized power iteration（仅考虑 real case）

1. Rayleigh quotient：$x\cdot\lambda\approx Ax\Rightarrow\min_\lambda\|x\lambda-Ax\|_2^2$

$$\Rightarrow\lambda=\frac{x^TAx}{x^Tx}=\frac{x^TAx}{\|x\|^2}=:r(x)\quad\Big(\frac{x^HAx}{x^Hx}=\frac{x^HAx}{\|x\|^2}\Big)$$

2. Normalized Power iteration

```
input: A ∈ C^{n×n} and a starting point x^0 with ||x^0|| = 1.
for k = 1 to max
    x̃^k = A x^{k-1}.            (左乘A)
    x^k = x̃^k / ||x̃^k||.       (normalize x̃^k)
    σ_k = (x^k)^H A x^k.        (计算 Rayleigh quotient)(||x^k||=1)
end
output: x^k (or x^max).
```

① 每个 iteration 的 complexity 为 $O(n^2)$（A 为 sparse 则可以减少）

② 计算 σ_k 时的 Ax^k 的计算可被省略 ③ Convergence rate 取决于 $\left|\frac{\lambda_2}{\lambda_1}\right|$

3. Normalized power method 的 convergence

normalized power method 的每步 iterate 满足：

① $\|x^k-(\pm/e^{i\theta})v_1\|=O\big(\left|\frac{\lambda_2}{\lambda_1}\right|^k\big)$ ② $|\sigma_k-\lambda_1|=O\big(\left|\frac{\lambda_2}{\lambda_1}\right|^k\big)$

注：① 假设 eigenvectors v_i，$i=1,\cdots,n$ 为 normalized

② $\{x^k\}_k$ 不一定收敛，x^k 的 sign / phase 可能会 "jump"

③ 若 A 为 Hermitian，则有 $|\sigma_k-\lambda_1|=O\big(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\big)$

④ x^k 的 sign / phase 取决于 $\big(\frac{\lambda_1}{|\lambda_1|}\big)^k\cdot\frac{\alpha_1}{|\alpha_1|}=\text{sign}(\alpha_1)\cdot\text{sign}(\lambda_1)^k$

即同时取决于 x_0 和 v_1 的夹角与 λ_1 的符号

⑤ Normalized power method 更 stable 的原因：

· 避免 overflow 和 underflow · Rayleigh quotient 是更好的近似

4. Power iteration with shift

① 选取合适的 σ，将 A 化为 $A-\sigma I$ $\big(\left|\frac{\lambda_2-\sigma}{\lambda_1-\sigma}\right|<\left|\frac{\lambda_2}{\lambda_1}\right|\big)$

② 用 normalized power iteration 求出 $A-\sigma I$ 的 eigen-pair

③ 将 σ 加回求出的 $\lambda_1-\sigma$，得到 A 的一个 eigenvalue λ_1

注：若要提升最大，则应 $\min_\sigma\left|\frac{\lambda_2'}{\lambda_1-\sigma}\right|$

§8 Inverse iteration

1. inverse iteration 的 scheme

① $A\tilde{x}^k=x^{k-1}$ ($\tilde{x}^k=A^{-1}x^{k-1}$) ② $x^k=\frac{\tilde{x}^k}{\|\tilde{x}^k\|}$ ③ $\sigma_k=(x^k)^HAx^k$

注：① x^k 会收敛至对应A的 smallest eigenvalue 的 eigenvector

② Rayleigh quotient σ_k 会收敛至A的 smallest eigenvalue

2. Inverse iteration with shifts

```
input: A ∈ C^{n×n}, starting point x^0 with ||x^0|| = 1, shift σ ∈ C.
for k = 1 to max
    (A − σI) x̃^k = x^{k-1}.   (x^k = (A−σI)^{-1} x^{k-1})   // inverse iteration
    x^k = x̃^k / ||x̃^k||.                                    // normalize
    σ_k = (x^k)^H A x^k.                                     // Rayleigh quotient
end
output: x^k (or x^max).
```

① complexity：$A-\sigma I$ 的 Initial factorization：$O(n^3)$

每个 iteration 中的 complexity：$O(n^2)$

② Convergence rate：取决于 $\left|\frac{\lambda_j-\sigma}{\lambda_k-\sigma}\right|$

λ_j 为最接近 σ 的 eigenvalue，λ_k 为第二接近 σ 的 eigenvalue

§9 Rayleigh quotient iteration

1. Rayleigh quotient iteration 的 algorithm

```
input: A ∈ C^{n×n}, x^0 with ||x^0|| = 1, and σ_0 = (x^0)^H A x^0.
for k = 1 to max
    (A − σ_{k-1} I) x̃^k = x^{k-1}.     /* inverse iteration */
    x^k = x̃^k / ||x̃^k||.              /* normalization */
    σ_k = (x^k)^H A x^k.               /* Rayleigh quotient */
end
output: x^k (or x^max).
```

① 对 symmetric matrices effective

② 每次迭代均需对 $A-\sigma_{k-1}I$ 重新 factorization.

2. Rayleigh quotient iteration 的 convergence（A 为 Hermitian）

① 对于 almost every（不是所有）x^0，RQI 均会收敛至一个 eigen-pair

② 令 (v_j,λ_j) 为一个 A 的一个 eigen-pair. 若 x^0 足够接近 eigenvector v_j，则

· $\|x^{k+1}-(\pm/e^{i\theta})v_j\|=O(\|x^k-(\pm/e^{i\theta})v_j\|^3)$

· $|\sigma_{k+1}-\lambda_j|=O(|\sigma_k-\lambda_j|^3)$ as $k\to\infty$

§10 (Pure) QR Iteration

1. QR Iteration 的 algorithm

```
input: Set X^0 = A.
for k = 1 to max
    Q_k R_k = X^{k-1}.    /* QR factorization of X^{k-1} */
    X^k = R_k Q_k.        /* Recombine factor in reverse order */
end
```

① 特定假设下，X^k 会收敛至 A 的 Schur factorization：$X^k\to T$，$A=QTQ^H$

② $\{X^k\}_k$ 彼此间 unitarily similar：$X^k=Q_k^HX^{k-1}Q_k=Q_k^H\cdots Q_1^HAQ_1\cdots Q_k$

③ 若 $X^0=A$ 为 symmetric，则 QR iteration 会保持 symmetry，$X^k\to$ diagonal

2. QR Iteration 的原理 Core procedure：重复以下步骤：

(1) Solve $(A-\sigma I)\tilde{v}=e_n$ (3) 构造 orthorgonal matrix：$Q=[\tilde{Q}\ v]$

(2) Normalize：$v=\hat{v}/\|\tilde{v}\|$ (4) Update A：$A=Q^TAQ$

one step inverse power iteration

① 上述过程会对 $Q_1\cdots Q_k$ 的最后一列进行 Inverse iteration with shifts

② 上述过程会将 A 的最后一行和最后一列化为 diagonal form.

3. (Pure) QR algorithm 的 convergence

若 A 的某些 eigenvalues 有 same modulus，则该方法不一定收敛.

§11 QR algorithm with shift

1. QR algorithm with shift

```
input: Set X^0 = A.
for k = 1 to max
    Pick a shift σ_{k-1}.
    Q_k R_k = X^{k-1} − σ_{k-1} I.   /* QR factorization */
    X^k = R_k Q_k + σ_{k-1} I.       /* Recombine factors */
end
```

2. σ 的选取：Rayleigh quotient shift

$$\sigma_k=\frac{(\hat{q}^k)^TA\hat{q}^k}{\|\hat{q}^k\|^2}\ (\hat{q}^k\ 为\ \hat{Q}_k=Q_1\cdots Q_k\ 的最后一列)=(\hat{q}^k)^TA\hat{q}^k$$

$=e_n^T\hat{Q}_k^TA\hat{Q}_ke_n=e_n^TX^ke_n$ ($\hat{Q}_k^TA\hat{Q}_k=X^k$) $=X_{nn}^k$

收敛速度：cubic local convergence

缺点：对于某些 initial points 可能会 fail（如 $C=[0,1;1,0]$）

3. σ 的选取：Wilkinson shift

令 $X=\begin{bmatrix}x_{n-1} & y_{n-1}\\ y_{n-1} & x_n\end{bmatrix}$ Wilkinson shift σ 为 submatrix X 中接近 x_n 的 eigenvalue

收敛速度：cubic convergence（worst case 为 quadratic convergence）

优点：若 input 矩阵 A 为 real symmetric & tridiagonal，则一定收敛

4. Deflation 若 $A\in R^{n\times n}$ 为 symmetric，X^k 为 QR algorithm 的 k-th iteration 的结果

$$X^k=\hat{Q}_k^TA\hat{Q}_k=\begin{bmatrix}B & r_n\\ r_n^T & X_{nn}^k\end{bmatrix}\approx\begin{bmatrix}B & 0\\ 0^T & \lambda_n\end{bmatrix}$$

① 基本思路：· 若 $\|r_n\|<tol$，则停止该 iteration，$X_{nn}^k\approx\lambda_n$

· 接下来对 smaller matrix B 进行 QR algorithm

$$Y^k=\hat{U}_k^TB\hat{U}_k=\begin{bmatrix}C & r_{n-1}\\ r_{n-1}^T & Y_{n-1,n-1}^k\end{bmatrix}\approx\begin{bmatrix}C & 0\\ 0^T & \lambda_{n-1}\end{bmatrix}$$ 相当于

$$\begin{bmatrix}\hat{U}_k^T & 0\\ 0^T & 1\end{bmatrix}\hat{Q}_k^TA\hat{Q}_k\begin{bmatrix}\hat{U}_k & 0\\ 0^T & 1\end{bmatrix}\approx\begin{bmatrix}C & 0 & 0\\ 0^T & \lambda_{n-1} & 0\\ 0^T & 0 & \lambda_n\end{bmatrix}$$

· A 的 original eigenvectors 可以通过将 $\hat{Q}_k$，$\hat{U}_k$，… explicitly 乘起来得到

注：① 该思路同样适用于 nonsymmetric case，但结果矩阵为上三角矩阵

② Flops：# flops：1^{st} deflation = $\underbrace{\#\ iter}_{\approx O(1)}$ · # flops for one QR step $=O(n^3)$

总 cost：total costs $=O(n^3+(n-1)^3+\cdots+1)=O(n^4)$

§12 The reduction phase（QR iteration 的两个 phases）

1. Hessenberg matrix 在 triangular 上额外多出了一个与主对角线相邻的 diagonal

① 任意矩阵均可以通过 orthorgonal transformation 在有限步内化为 Hessenberg

② Symmetric Hessenberg matrix 为 tridiagonal

2. Reduction phase 的步骤

Step 1：将 a_{21} 以下的 entries 化为 0：

· 若 $a_1=[a_{11}\ a^T]^T$ 为 A 的第一列，令

$Q_1^T=\begin{bmatrix}1 & 0^T\\ 0 & H_{v_1}\end{bmatrix}$，其中 $H_{v_1}=I-\frac{2v_1v_1^T}{\|v_1\|^2}$，$v_1=a-\alpha_1e_1$，$\alpha_1=\pm\|a\|_2$

$Q_1^Ta_1=\begin{bmatrix}a_{11}\\ H_{v_1}a\end{bmatrix}=\begin{bmatrix}a_{11}\\ \alpha_1\\ 0\end{bmatrix}$

· 为了不改变 eigenvalues，再对 A 右乘 Q_1（Q^TAQ 为相似变换）

注：full reduction algorithm 的 cost：（最后化为对称阵）

· 一般情况：$\approx\frac{10}{3}n^3$ flops · Symmetric case：$\approx\frac{4}{3}n^3$ flops

3. Reduction phase 能减少 computational cost 的原理

① 若 $H=QR$ 为 upper Hessenberg，则 Q 和 RQ 也为 upper Hessenberg

⇒ 可以使用更 cheaper 的方式进行 QR factorization（如 Givens rotations）

② Cost：· General matrices：$O(n^3)\to O(n^2)$ · Symmetric matrices：$\to O(n)$

4. QR iteration：two phase 总 cost

real symmetric：$\frac{4}{3}n^3/9n^3$；general matrices：$10n^3/25n^3$

8. Singular value decomposition

§1 Singular value decomposition

1. SVD：一个 real $m\times n$ 矩阵 A：$A=U\Sigma V^T$

① $U\in R^{m\times m}$ 的 columns u_i 被称为 left singular vectors

② $V\in R^{n\times n}$ 的 columns v_i 被称为 right singular vectors

③ $\Sigma\in R^{m\times n}$ 为一 diagonal matrix with $\sigma_{ii}=\sigma_i\ge0$ 为 A 的 singular values

且通常为 ordered：$\sigma_1\ge\sigma_2\ge\cdots\ge\sigma_p\ge0$，$p=\min\{m,n\}$

注：① 若 $U=V$，则 SVD 等价于 eigendecomposition（仅仅 $m=n$ 还不能得出该结论）

2、SVD 的 existence 与 construction

所有 real matrix $A\in R^{m\times n}$ 均有 SVD：$A=U\Sigma V^T$，且

① U 为 $AA^T\in R^{m\times m}$ 的 eigenvectors ② V 为 $A^TA\in R^{n\times n}$ 的 eigenvectors

③ singular values $\sigma_i=\sqrt{\lambda_i}$，$i=1,\cdots,p$，其中 $\lambda_1,\cdots,\lambda_p$ 为 AA^T 与 A^TA 中维度较小的的eigenvalues

3、thin SVD（Economy size）

r 为 matrix A 的非零 singular values 数：$\sigma_1\ge\cdots\ge\sigma_r>0=\sigma_{r+1}=\cdots=\sigma_p$

则 $A=[\underbrace{U_1}_{r}\ \underbrace{U_2}_{m-r}]\begin{array}{c}r\\m-r\end{array}\begin{bmatrix}\tilde\Sigma & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T\\V_2^T\end{bmatrix}\begin{array}{c}\}r\\ \}n-r\end{array}$ 则 thin SVD 为：$A=U_1\tilde\Sigma V_1^T$

4、对 Left & Right singular vectors 的理解

$$A=U\Sigma V^T\iff\begin{cases}AV=U\Sigma\\U^TA=\Sigma V^T\end{cases}\iff\begin{cases}Av_i=\sigma_i u_i\\u_i^TA=\sigma_i v_i\end{cases}$$

（v_i 为右奇异向量，类似右乘特征向量）（u_i 为左奇异向量，类似左乘特征向量）

5、计算SVD

$A=\begin{bmatrix}1&0&-1\\1&1&0\end{bmatrix}$ ① Step 1：计算 A^TA 与 AA^T 中维度更小的一个 $AA^T=\begin{bmatrix}2&1\\1&2\end{bmatrix}$

② Step 2：计算 A^TA 或 AA^T 的特征值，求出 Σ

$\det(AA^T-\lambda I)=0\Rightarrow\lambda_1=3,\ \lambda_2=1\Rightarrow\sigma_1=\sqrt3,\ \sigma_2=1\Rightarrow\Sigma=\begin{bmatrix}\sqrt3&0\\0&1\end{bmatrix}$

③ Step 3：计算 A^TA 或 AA^T 的特征向量，求出 V 或 U

$AA^T-\lambda_1I=0\iff\left[\begin{array}{cc|c}1&-1&0\\0&0&0\end{array}\right]$ 取自由未知量 $x_2=1$，有 eigenvector $[1,1]^T$ Normalize 后有：$u_1=\frac{1}{\sqrt2}[1,1]^T$

$AA^T-\lambda_2I=0\iff\left[\begin{array}{cc|c}1&1&0\\0&0&0\end{array}\right]$ 取自由未知量 $x_2=1$，有 eigenvector $[-1,1]^T$ Normalize 后有：$u_2=\frac{1}{\sqrt2}[-1,1]^T$

$\Rightarrow U=\frac{1}{\sqrt2}\begin{bmatrix}1&-1\\1&1\end{bmatrix}$

④ Step 4：利用 $A=U\Sigma V^T$ 求解另一个 orthorgonal matrix

$A=U\Sigma V^T\iff A^T=V\Sigma U^T\iff V=A^TU\Sigma^{-1}$

$\Rightarrow V=\begin{bmatrix}1&1\\0&1\\-1&0\end{bmatrix}\frac{1}{\sqrt2}\begin{bmatrix}1&-1\\1&1\end{bmatrix}\begin{bmatrix}\frac{\sqrt3}{3}&0\\0&1\end{bmatrix}=\frac{1}{\sqrt6}\begin{bmatrix}2&0\\1&\sqrt3\\-1&\sqrt3\end{bmatrix}$

§2 SVD 的应用：Low-rank matrix approximation

1、利用 SVD 改写 A：$A=\sum_{i=1}^{r}\sigma_i u_i v_i^T$

2、Best Low-rank approximation 若 A 有 rank r，且给定 $0\le k\le r$，则

$\|A-A_{(k)}\|_F^2=\min_{\mathrm{rank}(X)\le k}\|X-A\|_F^2=\sigma_{k+1}^2+\cdots+\sigma_r^2$

即 $A_{(k)}=\sum_{i=1}^{k}\sigma_iS_i=\sum_{i=1}^{k}\sigma_iu_iv_i^T$ 为限定 $\mathrm{rank}(X)\le k$ 时的 best approximation

§3 计算 SVD 的 algorithm

1、第一种想法：直接计算 $A=U\Sigma V^T$：会导致 loss of precision

2、第二种想法：将 SVD 化为 eigenvalue problem 给定 $A\in R^{m\times n}$（假设 $m>n$），

则 $B=\begin{array}{c}\begin{array}{cc}n&m\end{array}\\\begin{bmatrix}0&A^T\\A&0\end{bmatrix}\end{array}\begin{array}{c}n\\m\end{array}$ 为symmetric，由 $A=U_1\Sigma V^T$，$U=[\overset{n}{U_1}\ \overset{m-n}{U_2}]$ 有

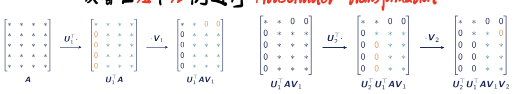
$P^TBP=\begin{bmatrix}\Sigma&0&0\\0&0&0\\0&0&-\Sigma\end{bmatrix}$ 其中 $P=\frac{1}{\sqrt2}\begin{bmatrix}V&0&V\\U_1&\sqrt2U_2&-U_1\end{bmatrix}$

因此对 B 进行 Power iteration 或 QR algorithm

3、第三种想法：Two-phase approach

Phase I：Golub-Kahan bidiagonalization

交替在左和右侧进行 Householder transformation



- Let $b^\top=[b_1\ \ b_2\ \ b_3\ \ b_4]=[b_1\ \ \beta^\top]$ be the first row of $U_1^\top A$.
- We can build V_1 as follows:

$$v_1=\begin{bmatrix}b_2\\b_3\\b_4\end{bmatrix}\pm\|\beta\|\begin{bmatrix}1\\0\\0\end{bmatrix},\quad H_{v_1}=I-\frac{2v_1v_1^\top}{\|v_1\|^2},\quad V_1=\begin{bmatrix}1&0\\0&H_{v_1}\end{bmatrix}.$$

- The total costs for Phase I are essentially identical to computing two QR factorizations of the matrix A: $\approx 4mn^2-\frac43n^3$ flops.
- There are also faster variants that requires fewer flops.

Phase II：使用 specialized QR algorithm 得到 diagonal form 与 singular values

§4 Minimum norm solutions 与 pseudoinverse

假设 $\sigma_1\ge\cdots\ge\sigma_r>0=\sigma_{r+1}=\cdots=\sigma_p$，有

$$A=[U_1\ U_2]\begin{bmatrix}\tilde\Sigma&0\\0&0\end{bmatrix}\begin{bmatrix}V_1^T\\V_2^T\end{bmatrix}=U_1\tilde\Sigma V_1^T=\sum_{i=1}^{r}\sigma_iu_iv_i^T$$

1. Orthorgonal basis

① U_1 的 columns 为 $\mathrm{range}(A)$ 的 orthonormal basis（即 $\mathrm{range}(U_1)=\mathrm{range}(A)$）

② U_2 的 columns 为 $\mathrm{range}(A)^\perp$ 的 orthonormal basis（由 $U_1^TU_2=0$ 易得）

③ V_2 的 columns 为 $\mathrm{null}(A)$ 的 orthonormal basis，即 $\mathrm{range}(V_2)=\mathrm{null}(A)$

证 ①（仅证明 $x\in\mathrm{Range}(A)\Rightarrow x\in\mathrm{Range}(U_1)$）

$x\in\mathrm{Range}(A)\Rightarrow\exists y$ s.t. $x=Ay\Rightarrow x=\sum_{i=1}^{r}\sigma_iu_iv_i^Ty$

$=\sum_{i=1}^{r}\sigma_i(v_i^Ty)u_i\Rightarrow x\in\mathrm{Range}(U_1)$

证 ③（仅证明 $x\in\mathrm{Range}(V_2)\Rightarrow x\in\mathrm{null}(A)$）

$x\in\mathrm{Range}(V_2)\Rightarrow\exists y$ s.t. $x=V_2y\Rightarrow Ax=U_1\tilde\Sigma V_1^Tx=$

$U_1\tilde\Sigma V_1^TV_2y=0$（$V_1^TV_2=0$）$\Rightarrow x\in\mathrm{null}(A)$

2、利用 SVD 解 linear system（$A\in R^{m\times n}$，$b\in R^m$，求 $Ax=b$ 的解）

① 由 SVD：$A=U_1\tilde\Sigma V_1^T$，有 $Ax=b\iff U_1\tilde\Sigma V_1^Tx=b$

② 由 $U_1^TU_1=I$，有 $U_1\tilde\Sigma V_1^Tx=b\iff V_1^Tx=\tilde\Sigma^{-1}U_1^Tb$

由 $U_2^TU_1=0$，有 $U_1\tilde\Sigma V_1^Tx=b\Rightarrow U_2^Tb=0$

（$\iff b\in\mathrm{range}(U_2^T)\iff b\in\mathrm{range}(U_2)^\perp\iff b\in\mathrm{range}(A)$）

③ 由 $V_1V_1^T=I-V_2V_2^T$，所有 $Ax=b$ 的解 x^* 均满足

$V_1V_1^Tx=V_1\tilde\Sigma^{-1}U_1^Tb\iff x=V_1\tilde\Sigma^{-1}U_1^Tb+\eta$，where $\eta\in\mathrm{Null}(A)$

注：若 A 有 full rank $r=n$，则不存在 V_2，此时 $Ax=b$ 有唯一解

$x^*=V_1\tilde\Sigma^{-1}U_1^Tb$（若 $b\in\mathrm{range}(A)$）

3、Minimum norm solution

对于 minimum norm problem：$\min_{x\in R^n}\|x\|_2^2$ s.t. $Ax=b$

令 $x^+:=V_1\tilde\Sigma^{-1}U_1^Tb$，则 $Ax=b$ 的解为 $x=x^++\eta$，$\eta\in\mathrm{Null}(A)$

$\Rightarrow\|x\|_2^2=\|x^+\|_2^2+2\underbrace{\eta}_{\in\mathrm{range}(V_2)}{}^T\underbrace{x^+}_{\in\mathrm{range}(V_1)}+\|\eta\|_2^2=\|x^+\|_2^2+\|\eta\|_2^2$

$\Rightarrow x^+$ 为 the unique min. norm solution

4、Pseudoinverse：$A^+:=V_1\tilde\Sigma^{-1}U_1^T$

① 若 A 为 square 且 nonsingular，则 $A^+=A^{-1}$

② 若 A 有 full column rank $r=n$，则 $A^+=(A^TA)^{-1}A^T$

③ $Ax=b$ 的 minimum norm solution 为 $x^+=A^+b$ $\quad x^+=A^+b$

④ Least square problem $\min_x\|Ax-b\|^2$ 的minimum norm solution为

（$\iff\min_x\|x\|^2$ s.t. $A^TAx=A^Tb$）

9. Iterative solver

§1 The conjugate direction method 解 $Ax=b$，A symmetric 且 $A\succ0$

1. Background：问题可被转化为 $\min_{x\in R^n}\phi(x)=\frac12x^TAx-b^Tx$

关注 residuals $r^k:=Ax^k-b=\nabla\phi(x^k)$ 的 convergence

2、Conjugate vectors：$\{p^0,p^1,\cdots,p^l\}$ 被称为 conjugate w.r.t. $A\succ0$，若 $(p^i)^TAp^j=0,\ \forall i\ne j$ ① orthogonal 可视作 conjugate 的特例（$A=I$）

② $p^0,p^1,\cdots,p^l$ 线性无关 ③ A 的 eigenvectors 为一组 conjugate vectors

3、Conjugate direction method 给定 $x^0\in R^n$ 与 conjugate directions $\{p^0,p^1,\cdots,p^{n-1}\}$

$x^{k+1}=x^k+\alpha_kp^k$，$\alpha_k=-\frac{(r^k)^Tp^k}{(p^k)^TAp^k}$（遵循了 exact line search $\min_\alpha\phi(x^k+\alpha_kp^k):=h(\alpha)$）

4. Conjugate directions 与 convergence

$\forall x^0\in R^n$，至多需要 n steps 找到 $Ax^*=b$ 的解 x^*

§2 The conjugate gradient method

1. The conjugate gradient method：p^k 被选为 $-r^k$ 和 p^{k-1} 的线性组合：

$p^k=-r^k+\beta_kp^{k-1}$，$\beta_k=\frac{(r^k)^TAp^{k-1}}{(p^{k-1})^TAp^{k-1}}$ → s.t. $(p^{k-1})^TAp^k=0$，$r^k:=Ax^k-b$

2、Algorithm：The conjugate gradient method

1. Initialization: Select an initial point $x^0\in\mathbb{R}^n$ and set $\texttt{tol}\ge0$, $r^0=Ax^0-b$, and $p^0=-r^0$.

For $k=0,1,\ldots$: $\quad r^{k+1}=Ax^{k+1}-b=Ax^k+\alpha_kAp^k-b$

2. Compute the updates $\boxed{\alpha_k=-\frac{(r^k)^\top p^k}{(p^k)^\top Ap^k}}$ $\quad=r^k+\alpha_kAp^k$

$x^{k+1}=x^k+\alpha_kp^k$ and $r^{k+1}=r^k+\alpha_kAp^k$.*

3. If $\|r^{k+1}\|\le\texttt{tol}$, then STOP and x^{k+1} is the output.

4. Calculate $\boxed{\beta_{k+1}=\frac{(r^{k+1})^\top Ap^k}{(p^k)^\top Ap^k}}$ and $p^{k+1}=-r^{k+1}+\beta_{k+1}p^k$. $\quad i<k$

3、conjugate directions：$(p^k)^TAp^i=0$，$(r^k)^Tr^i=0$，$(r^k)^Tp^i=0$

4、convergence rate：$\|x^k-x^*\|_A\le2\left[\frac{\sqrt K-1}{\sqrt K+1}\right]^k\|x^0-x^*\|_A$（linear）

（$\|x\|_A^2=x^TAx$（A-norm））

5、It is possible to simplify the update rules in the CG-method to:

$-(r^k)^Tp^k=-(r^k)^T(-r^k+\beta_kp^{k-1})=\|r^k\|^2$

☆ $\boxed{\alpha_k=\frac{\|r^k\|^2}{(p^k)^\top Ap^k}\quad\text{and}\quad\beta_{k+1}=\frac{\|r^{k+1}\|^2}{\|r^k\|^2}.}$

10. Nonlinear equations

§1 Nonlinear equations（$F:R^n\to R^n$，求 $F(x)=0$ 的 roots $x\in R^n$）

1. Existence & uniqueness：$f:R\to R$：利用 intermediate value theo

F：high-dim：任意数量的 solutions

2、Local uniqueness：① $F:R^n\to R^n$ continuously differentiable

② $x^*\in R^n$ 有 $F(x^*)=0$ ③ Jacobian $DF(x^*)$ nonsingular，则

x^* 为 local unique：$\exists$ nbhd $U\subset R^n$ of x^* s.t. $F(x)\ne0$ for $\forall x\in U\setminus\{x^*\}$.

$$DF(x)=\begin{bmatrix}\nabla F_1(x)^T\\\vdots\\\nabla F_n(x)^T\end{bmatrix}=\begin{bmatrix}\frac{\partial}{\partial x_1}F_1(x)&\cdots&\frac{\partial}{\partial x_n}F_1(x)\\\vdots&\ddots&\vdots\\\frac{\partial}{\partial x_1}F_n(x)&\cdots&\frac{\partial}{\partial x_n}F_n(x)\end{bmatrix}$$

§2 Conditioning $\quad f(x^*)=0\iff x^*=f^{-1}(0)$

1. relative condition number：$\mathrm{cond}(f)=\left|\frac{xf'(x)}{f(x)}\right|\Rightarrow\mathrm{cond}(f^{-1})=\left|\frac{f(x)}{xf'(x)}\right|$

2、absolute condition number：$\mathrm{cond}_{abs}(f)=\frac{|f(x+\Delta x)-f(x)|}{|\Delta x|}\approx|f'(x)|$

$\mathrm{cond}_{abs}(f^{-1})\approx\frac{1}{|f'(x)|}$ 若 $f'(x)\approx0$，则为 ill-conditioned

3. High-dimensional case：① $\mathrm{cond}(F)=\frac{\|x\|\cdot\|DF(x)\|}{\|F(x)\|}$

② $\mathrm{cond}_{abs}(F)=\|DF(x^*)^{-1}\|$ for root finding problem

4. Approximate solutions：$\|F(\hat x)\|\approx0$（不是 $\|x-x^*\|\approx0$）

§3 Bisection method

1. Bisection method

local minimizer 必须满足 FONC：$f'(x)=0\Rightarrow g(x)=f'(x)$ 的求根问题

若有两点 x_l 与 x_r 满足 $g(x_l)<0$，$g(x_r)>0$，则根据介值定理，

若 $g(\cdot)$ 连续，则一定存在一个 $g(\cdot)$ 的根落在 $[x_l,x_r]$ 内

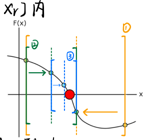
① 定义 $x_m=\frac{x_l+x_r}{2}$

② 若 $g(x_m)=0$，则 output x_m

③ 否则 · 若 $g(x_m)>0$，则令 $x_r=x_m$

· 若 $g(x_m)<0$，则令 $x_l=x_m$

④ 若 $|x_r-x_l|<\varepsilon$，则终止并输出 $\frac{x_l+x_r}{2}$，否则回到 step ①



2、对 bisection method 的分析

① bisection method 每次将区间长度减小一半（linear）

② 若要求出 x^* 的一个 ε approximation，则至多进行 $\log_2\frac{x_r-x_l}{\varepsilon}$ 次迭代

③ 对 $f'(x)$ 使用 bisection method，可求出满足 FONC（approximately）的点。若 $f(x)$ convex，则可求出 $f(x)$ 的 global minimizer（approximately）

§4 Fixed-point iteration（1D）

1. Fixed-point problem：求 $x=g(x)$ 的 x，用 fixed-point iteration 求：

$x^{k+1}=g(x^k)$（可以将 $f(x)=0$ 转化为 $x=g(x)$）

2、Lipschitz continuity

A function $g:\mathbb{R}\to\mathbb{R}$ is Lipschitz continuous with constant $L\ge0$ if

$$|g(x)-g(y)|\le L\cdot|x-y|\quad\forall\,x,y\in\mathbb{R}.\iff|g'(x)|\le L\quad\forall x\in R$$

If $L<1$, then g is called a contractive mapping or contractive.

3. Fixed-point method 的 convergence

Let $x^*=g(x^*)$ be a fixed-point of g and suppose that there is a neighborhood (interval) U containing x^* such that g is contractive on U with constant $L<1$. $\iff|g'(x)|\le L,\ \forall x\in U$

则 $x^{k+1}=g(x^k)$ 以 rate L linear converges to x^*（$\forall x^0\in U$）

① $|g'(x^*)|<1\Rightarrow|g'(x)|<1$，$\forall x\in N_\delta(x^*)$

② 若 $g'(x^*)=0$ 且 g 为 2阶连续可微 ⇒ convergence rate 至少 quadratic

$|x^{k+1}-x^*|=|g(x^k)-g(x^*)|=|g'(x^*)(x^k-x^*)+O(|x^k-x^*|^2)|=O(|x^k-x^*|^2)$

③ 若 $|g'(x^*)|>1$，则不能收敛到 x^* / g 有 unique fixed-point in U

4、Existence：若 g 在 U 上 contractive，且 $g:U\to U$（$g(U)\subseteq U$），则

§5. Newton's method（1D）

1、Newton's method 的特征：① 收敛速度更快 ② 需要二阶导的信息 ③ sensitive to initial point

2、Newton's method 的基本思路：用 x^k 处的 first-order Taylor expansion 来近似任意点 $g(x)$

$g(x)\approx g(x^k)+g'(x^k)(x-x^k)$ 令右侧等于0，即可解出 x^{k+1}：$x^{k+1}=x^k-\frac{g(x^k)}{g'(x^k)}$

注：① 对于某些 initial point，Newton's method 不一定收敛 ② convex 不能确保收敛

3、Newton's method 的收敛模式

① 若 $g(x)$ 为 twice continuously differentiable，且 x^* 为 $g(x)$ 的根，$g'(x^*)\ne0$。

则当 $|x^0-x^*|$ 足够小（确保能收敛）时，Newton iterations 生成的 sequence 满足

$|x^{k+1}-x^*|\le C|x^k-x^*|^2$ 其中 $C=\sup_x\frac12\left|\frac{g''(x)}{g'(x)}\right|$（取决于 curvature），

② 我们称这种 convergence speed 为 quadratic convergence

4、与 fixed-point iter 的联系：$f(x)=0\iff x=g(x):=x-\frac{f(x)}{f'(x)}$

因此有 $g'(x)=\frac{f(x)f''(x)}{(f'(x))^2}$，若 $f(x^*)=0$ 且 $f'(x^*)\ne0$，则 $g'(x^*)=0$（二次收敛）

§6 Nonlinear system 的 algorithm：fixed-point iteration

1. Fixed-point problem：求 $x=G(x)$ 的 x，用 fixed-point iteration 求：

$x^{k+1}=G(x^k)$（可以将 $F(x)=0$ 转化为 $x=G(x)$）

2、Lipschitz continuity：

- Lipschitz continuity can be generalized to n-dimensional mappings.
- G is Lipschitz continuous (on $\mathbb{R}^n$) with constant $L\ge0$ if

判断时可利用各norm equivalent $\|G(x)-G(y)\|_2\le L\cdot\|x-y\|_2\quad\forall\,x,y\in\mathbb{R}^n$.

- If $L<1$, then G is said to be contractive.
- The mapping G is Lipschitz continuous on a convex set $U\subset\mathbb{R}^n$ if and only if $\|DG(x)\|_2\le L$ for all $x\in U$.

3、Fixed-point method 的 convergence

- Conditions for convergence are similar to the 1D case.
- If G is contractive on U, then the fixed-point scheme $x^{k+1}=G(x^k)$ converges as long as $x^0\in U$.
- The convergence is typically linear with rate $\eta\approx\|DG(x^*)\|$.
- If $DG(x^*)=0$, then the convergence is quadratic.

§7 Nonlinear system 的 algorithm：Newton's method

1、形式：$x^{k+1}=x^k-DF(x^k)^{-1}F(x^k)$

2、实际实现：$DF(x^k)d^k=-F(x^k)\Rightarrow x^{k+1}=x^k+d^k$

3、convergence：quadratic（但起点要 sufficiently close to x^*）

4、Cost：算 DF：n^2，解 linear system：$O(n^3)$

§8 Globalized Newton's method

1、step size 的选取：① merit function：$\min_x m(x)=\frac12\|F(x)\|^2$

Observations:

- x^* solves $F(x^*)=0\iff m(x^*)=0$, i.e., x^* is a global solution of problem (1).
- It holds that $\nabla m(x)=DF(x)^\top F(x)$.

⇝ If $DF(x)$ is invertible (for all x), then stationary points of m need to satisfy $F(x)=0$.

- Let d be the Newton direction with $DF(x)d=-F(x)$, then:

$$\nabla m(x)^\top d=F(x)^\top DF(x)d=-\|F(x)\|^2<0.$$

⇝ The Newton direction is always a descent direction for m!

The Globalized Newton Method

1. Initialization: Choose an initial point $x^0\in\mathbb{R}^n$.

For $k=0,1,2,\ldots$:

2. STOP, if $\|\nabla m(x^k)\|\le\texttt{tol}$.
3. Compute d^k by solving the Newton equation $DF(x^k)d^k=-F(x^k)$.
4. Perform backtracking on the merit function m to calculate a step size α_k. Set $x^{k+1}=x^k+\alpha_kd^k$.

Comments:

- This is a full analog of the globalized Newton method for optimization problems: $m\rightsquigarrow f$.
- Locally, under certain assumptions, the algorithm turns into a pure Newton method with $\alpha_k=1$ for all k sufficiently large.

⇝ Fast local (quadratic) convergence of $\{x^k\}_k$ can be observed.

2、α 的选取（backtracking line search）（回溯线搜索）

假设 d^k 已知，希望求解 α_k，则遵循以下步骤：

① 选取一个较小的常数 $\alpha\in(0,0.5)$，选取另一个常数 $0<\beta<1$ ② 令 $t=1$

③ 若 $f(x^k+td^k)\le f(x^k)+\alpha t\nabla f(x^k)^Td^k$，则选取 $\alpha_k=t$（Armijo condition）

否则令 $t=\beta t$ 并重复这一步骤

3、KKT conditions 若 x 为 local minimizer，则一定存在 λ,ν 与 η 使以下 conditions 满足：

Lagrangian 为：$L(x,\lambda,\nu,\eta)=f(x)+\sum_{i=1}^{m}\lambda_ig_i(x)+\sum_{i=1}^{p}\nu_ih_i(x)+\sum_{i=1}^{r}\eta_il_i(x)$

① Main condition

$D_jf(x)+\sum_{i=1}^{m}\lambda_iD_jg_i(x)+\sum_{i=1}^{p}\nu_iD_jh_i(x)+\sum_{i=1}^{r}\eta_iD_jl_i(x)\begin{cases}\ge0 & 若\ x_j\ge0\\\le0 & 若\ x_j\le0\\=0 & 若\ x_j\ \text{free}\end{cases}$

② Dual feasibility

$\lambda_i\le0\quad i=1,\cdots,m$（对应 $g_i(x)\ge0$）

ν_i free $\quad i=1,\cdots,p$（对应 $h_i(x)=0$）

$\eta_i\ge0\quad i=1,\cdots,r$（对应 $l_i(x)\le0$）

③ Complementarity

$\lambda_i\cdot g_i(x)=0\quad\forall i$

$\nu_i\cdot h_i(x)=0\quad\forall i$

$\eta_i\cdot l_i(x)=0\quad\forall i$

对应 Lagrangian 中添加的各项为0

$x_i\cdot D_{x_i}L(x,\lambda,\nu,\eta)=0\quad\forall i$（若 x_i free 则忽略）

④ Primal feasibility

$g_i(x)\ge0$，$h_i(x)=0$，$l_i(x)\le0$，$\forall i$

$x_i\ge0,\ \forall i\in M$；$x_i\le0,\ \forall i\in N$；x_i free，$\forall i\notin M\cup N$