高等数值算法与应用（三）

Advanced Numerical Algorithms & Applications

计算机科学与技术系 喻文健

Lecture 2

- Matrix condition number
- Relative residual, stable algorithm
- Permutation matrix, Elementary elimination matrix
- LU factorization, Partial pivoting
- Backward error estimation (growth factor)
- Computational complexity
- Rank-1 modification of matrix
- Rescaling, iterative refinement
- Cholesky factorization
- Solution of symmetric indefinite system

$$\frac{\|r\|}{\|A\|\|\hat{x}\|} \le \frac{\|E\|}{\|A\|}$$ 后验分析

How about their relationship? ⇒ or ⇐

$$LU = PA$$

$$\frac{\|E\|}{\|A\|} \overset{?}{\le} \rho n \varepsilon_{\text{mach}}$$

$$PAP^T = LDL^T$$

内容概要

- 矩阵分解及其应用
  - 六大分解简介
  - 矩阵与线性方程组求解基本理论
  - **LU**分解, **Cholesky**分解及其应用
  - **QR**分解与线性最小二乘问题
  - 特征值分解、奇异值分解(**SVD**)及其应用
  - 稀疏矩阵的直接解法

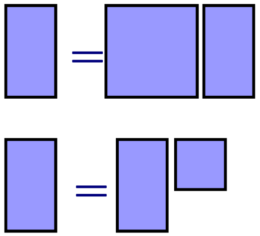
QR Factorization and Orthogonal Transformations

Wenjian Yu

- 本节主要内容
  - **QR**分解方法
  - 如何做**QR**分解：**Householder**变换
  - **Givens**旋转
  - **Gram-Schmidt**正交化
  - 小结

QR Factorization

$$A = QR$$



To computer QR factorization of *mxn* matrix ***A***, with $m \geq n$, annihilate subdiagonal entries of successive columns of ***A***, eventually reaching upper triangular form

Similar to LU factorization by Gaussian elimination, but uses orthogonal transformations instead of elementary elimination matrices

Possible methods include

$$Q^T A = R$$

- Householder transformations
- Givens rotations
- Gram-Schmidt orthogonalization

正交矩阵：

1.由单位正交基向量组成**(2-**范数**,**内积范数意义下**)**

2. $Q \cdot Q^{\mathrm{T}} = I$

类似作**LU**分解，要消去对角线下面的元素，但采用正交变换

Householder Transformation

将v乘上一个标量不影响**H**的结果

Householder transformation has form

$$H = I - 2\frac{vv^T}{v^T v}$$

for nonzero vector v

初等阵
Elementary matrix:
$$I - uv^T$$
Condition $v^T u \neq 1$ ensures nonsingular

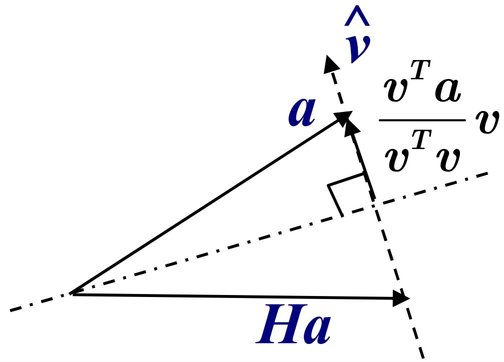
练习题2.26

$H = H^T = H^{-1}$, so H is orthogonal and symmetric

$$Ha = a - 2\frac{v^T a}{v^T v}v$$

Given vector a, we can choose v so that

$$Ha = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \alpha e_1$$



镜像变换

正交变换的性质

where $\alpha = \pm\|a\|_2$. The choice of v is

$$v = a - \alpha e_1$$

The sign of α is chosen to avoid cancellation

Example: Householder Transformation

Let $\boldsymbol{a} = [2 \;\; 1 \;\; 2]^T$

By foregoing recipe,

$$v = a - \alpha e_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \alpha \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} \alpha \\ 0 \\ 0 \end{bmatrix}$$

where $\alpha = \pm \|a\|_2 = \pm 3$

Since a_1 positive, choosing negative sign for α avoids cancellation

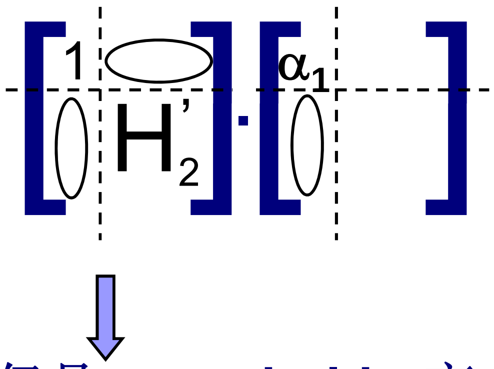
Thus, $v = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$

To confirm that transformation works,

$$Ha = a - 2\frac{v^T a}{v^T v} v = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 2\frac{15}{30}\begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$$

$H_2 \cdot H_1 A$: ***A*: mxn**

Householder QR Factorization



仍是**Householder**变换，在**H$_2$'**对应的***v'***前补分量**0**得到**H$_2$**的***v***

$$H = I - 2\frac{vv^T}{v^T v}$$

To compute QR factorization of ***A***, use Householder transformations to annihilate subdiagonal entries of each successive column

Each Householder transformation applied to entire matrix, but does not affect prior columns, so zeros preserved

In applying Householder transformation ***H*** to arbitrary vector ***u***,

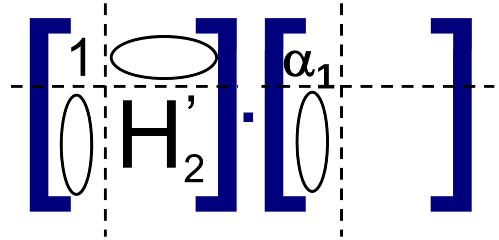
$$Hu = \left(I - 2\frac{vv^T}{v^T v} \right) u = u - 2\frac{v^T u}{v^T v} v$$

Which is much cheaper than general matrix-vector multiplication and requires only vector ***v***, not full matrix ***H***

有些应用是不需要得到**Q**矩阵的

Householder QR Factorization

$H_2 \cdot H_1 A$:   ***A*: mxn**



算法：

for k =1 to n

$\alpha_k = -\text{sign}(a_{kk})\sqrt{a_{kk}^2 + \cdots + a_{mk}^2}$

$v_k = [0, \ \cdots, \ 0, \ a_{kk}, \ \cdots, \ a_{mk}]^T - \alpha_k e_k$

$\beta_k = v_k^T v_k$

if $\beta_k = 0$ then   // $v_k = O$，说明对应的H_k=I

continue with next k

for j =k to n // $Ha_j = a_j - 2\dfrac{v^T a_j}{v^T v} v$

$\gamma_j = v_k^T a_j$

$a_j = a_j - (2\gamma_j / \beta_k) v_k$

end

end

存储：

v向量的非零元存在矩阵**A**的下三角部分

计算复杂度：

一个**m**维向量做**H**变换：**2m**的乘法计算量

$$\sum_{i=0}^{n-1} 2(m-i)(n-i) = mn^2 - n^3/3$$

2x2 Givens旋转变换矩阵

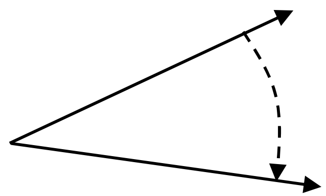
Givens rotations introduce zeros, one at a time

Given vector $[a_1\ a_2]^T$, choose scalars c and s so that

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

with $c^2 + s^2 = 1$, or equivalently, $\alpha = \sqrt{a_1^2 + a_2^2}$

顺时针旋转变换



Previous equation can be rewritten

$$\begin{bmatrix} a_1 & a_2 \\ a_2 & -a_1 \end{bmatrix}\begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

Gaussian elimination yields triangular system

$$\begin{bmatrix} a_1 & a_2 \\ 0 & -a_1 - a_2^2 / a_1 \end{bmatrix}\begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \alpha \\ -\alpha a_2 / a_1 \end{bmatrix}$$

Back-substitution then gives

$$s = \frac{\alpha a_2}{a_1^2 + a_2^2}, \qquad c = \frac{\alpha a_1}{a_1^2 + a_2^2}$$

Finally, $c^2 + s^2 = 1$ or $\alpha = \sqrt{a_1^2 + a_2^2}$, implies

$$s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}}, \qquad c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}}$$

Example:

Let $a = [4 \ \ 3]^T$ Compute cosine and sine,

$$s = \frac{a_2}{\sqrt{a_1^2 + a_2^2}} = \frac{3}{5} = 0.6, \qquad c = \frac{a_1}{\sqrt{a_1^2 + a_2^2}} = \frac{4}{5} = 0.8$$

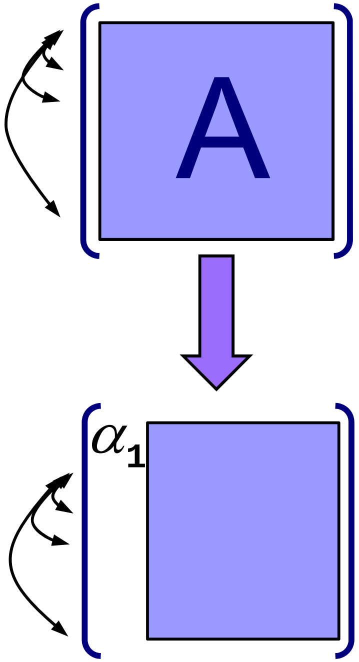
Rotation given by

$$G = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{bmatrix}$$

To confirm that rotation works,

$$Ga = \begin{bmatrix} 0.8 & 0.6 \\ -0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

Givens QR Factorization



一般的mxm Givens旋转矩阵，例如m=5:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & c & 0 & s & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} a_1 \\ \alpha \\ a_3 \\ 0 \\ a_5 \end{bmatrix}$$

旋转后的效果是将向量的第j个分量的值“添加”到第i个分量中。这个Givens旋转矩阵通过将2x2旋转阵“嵌入”到m阶单位阵的第i, j行和第i, j列中。

一个m维列向量做G变换：3m次乘法

Givens旋转矩阵与向量相乘，仅影响向量的两个分量，不会改变其他分量的值。

对稠密矩阵使用Givens旋转来正交三角化, 比使用Householder变换的计算量多50%, 并需要更多的存储空间, 因为每个旋转变换需两个参数c, s来定义

Householder方法：
存v向量中的浮点数占据下三角部分

Givens方法：
下三角部分每个需消去元素需存两个浮点数

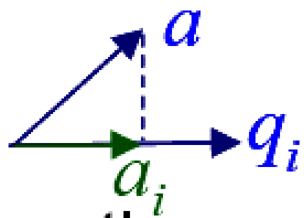
处理稀疏矩阵有优势！

Orthogonal components of a vector

单位正交基向量

- Given an *m*-vector a and an set of *n* orthonormal *m*-vectors $\{q_1, \dots q_n\}, n \le m$. Then a has a component a_i of length $\langle a, q_i \rangle$ along each direction of q_i: $a_i = \langle a, q_i \rangle q_i$

Define q_{n+1} as the "remainder" component:



$$q_{n+1} := a - (a_1 + \cdots + a_n)$$

- these *n+1* components $\{a_1, \dots, a_n, q_{n+1}\}$ form an orthogonal set

$$a = [q_1 \mid \cdots \mid q_n \mid q_{n+1}] \begin{bmatrix} \langle a, q_1 \rangle \\ \vdots \\ \langle a, q_n \rangle \\ 1 \end{bmatrix}$$

q_{n+1}与a_1, …, a_n都正交

- $q_{n+1} = 0$ if $a \in \text{span}(q_1, \dots q_n)$ or $m = n$

Gram-Schmidt orthogonalization and QR factorization

注意：列满秩的要求

- Given $A=[a_1,\ldots,a_n]\in\mathbb{R}^{m\times n}$ with full column rank. Consider the below procedure known as the Gram-Schmidt process
  - $v_1 := a_1,\ q_1 := v_1/\|v_1\|_2$
  - $v_2 := a_2 - \langle a_2,q_1\rangle q_1,\ q_2 := v_2/\|v_2\|_2$
  - $v_3 := a_3 - \langle a_3,q_1\rangle q_1 - \langle a_3,q_2\rangle q_2,\ q_3 := v_3/\|v_3\|_2$
  - $\cdots$
  - $v_n := a_n - \sum_{k=1}^{n-1}\langle a,q_k\rangle q_k,\ q_n := v_n/\|v_n\|_2$

A

$$\underbrace{[\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n]}_{\underset{m\times n}{A}} = [q_1,q_2,q_3,\ldots,q_n]\begin{bmatrix} \|v_1\|_2 & \langle a_2,q_1\rangle & \langle a_3,q_1\rangle & \cdots & \langle a_n,q_1\rangle \\ & \|v_2\|_2 & \langle a_3,q_2\rangle & \cdots & \langle a_n,q_2\rangle \\ & & \|v_3\|_2 & \cdots & \langle a_n,q_3\rangle \\ & & & \ddots & \vdots \\ & & & & \|\boldsymbol{v}_n\|_2 \end{bmatrix} =: \underset{m\times n}{Q}\ \underset{n\times n}{R}$$

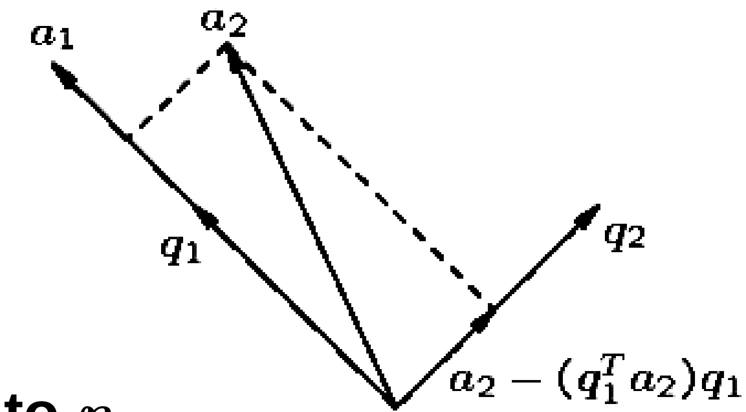
计算过程保证了{q_i}中任意两元素正交

Gram-Schmidt Orthogonalizatoin

Given vectors a_1 and a_2, can determine orthonormal vectors q_1 and q_2 with same span by orthogonalizing one vector against other:

注意与前面QR分解的区别：

$$\underset{m\times n}{A} = \underset{m\times n}{Q}\ \underset{n\times n}{R}$$



m>n时的简化QR分解怎么成为原始的QR分解？

for $k = 1$ **to** n

$\quad q_k = a_k$

$\quad$**for** $j = 1$ **to** k**-1**

$\quad\quad r_{jk} = q_j^T a_k$

$\quad\quad q_k = q_k - r_{jk} q_j$

$\quad$**end**

$\quad r_{kk} = \|q_k\|_2$

$\quad q_k = q_k / r_{jk}$

end

$$\sum_{k=1}^{n} 2m(k-1)$$

计算量mn^2

q_k为单位正交基

每次取一个列向量，让它与已生成的所有正交基进行正交化

单位化，normalization

Modified Gram-Schmidt

Classical Gram-Schmidt procedure often suffers loss of orthogonality in finite-precision

Also, separate storage is required for $\boldsymbol{A}$, $\boldsymbol{Q}$, and $\boldsymbol{R}$, since original $\boldsymbol{a}_k$ needed in inner loop, so $\boldsymbol{q}_k$ cannot overwrite columns of $\boldsymbol{A}$

$$\underset{m\times n}{\boldsymbol{A}} = \underset{m\times n}{\boldsymbol{Q}} \; \underset{n\times n}{\boldsymbol{R}}$$

Both deficiencies improved by modified Gram-Schmidt procedure, with each vector orthogonalized in turn against all subsequent vectors so $\boldsymbol{q}_k$ can overwrite $\boldsymbol{a}_k$:

for $k = 1$ **to** n

$r_{kk} = \|\boldsymbol{a}_k\|_2$

$\boldsymbol{q}_k = \boldsymbol{a}_k / r_{kk}$

for $j = k+1$ **to** n

$r_{kj} = \boldsymbol{q}_k^T \boldsymbol{a}_j$

$\boldsymbol{a}_j = \boldsymbol{a}_j - r_{kj}\boldsymbol{q}_k$

end

end

r_{ij}形成矩阵***R***，***Q***可存在***A***中

每生成一个正交向量，就与所有剩下的矩阵列向量进行正交化

“重正交技术” **reorthogonalization**

小结

- **Householder, Givens**变换的方法
  - 对应于$\underset{m\times n}{A} = \underset{m\times m}{Q}\ \underset{m\times n}{R}$ 的标准**QR**分解，算法不中断
  - 得到**R**矩阵，通过额外工作量可得到**Q**矩阵
  - 若要求**R**对角元为正，得到的**R**矩阵是唯一的 **??**
  - 对稠密阵**Householder**最有效，**Givens**适合于稀疏矩阵
- **G-S**正交化方法
  - 对应 $\underset{m\times n}{A} = \underset{m\times n}{Q}\ \underset{n\times n}{R}$ 的简化**QR**分解**(**当规定**R**对角元为正，分解唯一**)**，若矩阵**A**列不满秩，算法会中断
  - 得到**Q, R**矩阵**,** 经扩展可得标准**QR**分解；不适合稀疏矩阵
- 应用
  - **m=n**时，求解线性方程组，非常稳定$QRx = b \Rightarrow Rx = Q^T b$
  - 解线性最小二乘问题；构造一定特征值分布的矩阵

Application of QR Factorization – Linear Least Squares

Wenjian Yu

- 本节主要内容
  - 最小二乘问题
  - 线性最小二乘问题及其敏感性分析
  - 求解线性最小二乘问题的方法
    - 法方程法
    - 增广矩阵法
    - **QR**分解法
  - 列不满秩的情形，以及列选主元的**QR**分解

cencusgui.m 及思考题要求

超定方程组的求解问题

Linear system $\boldsymbol{Ax} = \boldsymbol{b}$, with *m*×*n* matrix A (*m*>*n*), is *overdetermined*

Least squares solution *x* minimizes squared Euclidean norm of residual vector $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{Ax}$, $\min_{\boldsymbol{x}} \|\boldsymbol{r}\|_2^2 = \min_{\boldsymbol{x}} \|\boldsymbol{b} - \boldsymbol{Ax}\|_2^2$

Problem noted by $\boldsymbol{Ax} \cong \boldsymbol{b}$, always has solution

Solution unique if, and only if, columns of ***A*** are linearly independent

$\text{rank}(\boldsymbol{A}) = \boldsymbol{n}$, where $\boldsymbol{A}$ is $\boldsymbol{m} \times \boldsymbol{n}$

假设列线性相关 → 解不唯一

Example:

Data Fitting

Given m data points (t_i, y_i), find n-vector x of parameters that gives "best fit" to model function $f(t, x)$:

$$\min_{\boldsymbol{x}} \sum_{i=1}^{m} (y_i - f(t_i, \boldsymbol{x}))^2$$

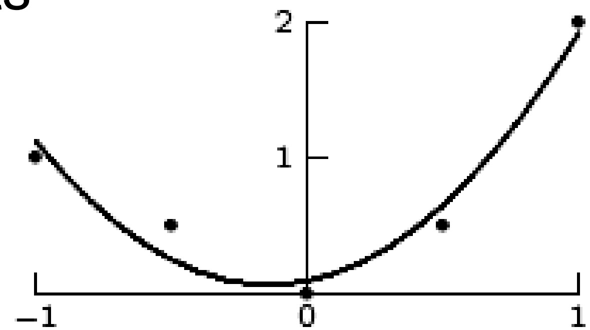
表格函数的最佳平方逼近

Problem *linear* if function *f* linear in components of x: $f(t_i, \boldsymbol{x}) = x_1\phi_1(t) + x_2\phi_2(t) + \cdots + x_n\phi_n(t)$

Where functions ϕ_j depend only on t

F对x来说是线性函数

Polynomial fitting $f(t, x) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1}$
is linear, since polynomial linear in coefficients



Fitting sum of exponentials, with
$f(t, x) = x_1 e^{x_2 t} + \cdots + x_{n-1} e^{x_n t}$, is nonlinear problem 非线性优化问题

Method for Linear Least Squares

To minimize $\|r\|_2^2 = r^T r = (b - Ax)^T (b - Ax)$
$= b^T b - 2x^T A^T b + x^T A^T A x$

take derivative with respect to x and set to 0, $2A^T A x - 2A^T b = 0$,
which reduces to $n \times n$ linear system $A^T A x = A^T b$ 达最小的充分条件：二阶偏导的Hessian矩阵正定
known as system of *normal equations* 法方程

Cholesky factorization $A^T A = L L^T$ can be used to obtain solution

法方程方法的推导

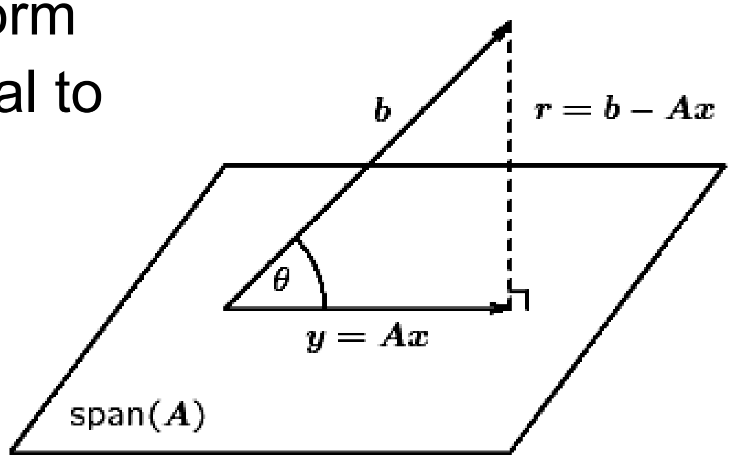
Vectors v_1 and v_2 are *orthogonal* if their inner product is zero, $v_1^T v_2 = \mathbf{0}$

另一种推导
geometric view

Space spanned by columns of $m \times n$ matrix A, span(A)= $\{Ax : x \in \mathbb{R}^n\}$, is of dimension at most n

If $m>n$, b generally does not lie in span(A), so no exact solution to $Ax=b$

Vector $y=Ax$ in span(A) closest to b in 2-norm occurs when residual $r = b - Ax$ orthogonal to span(A)



Thus, $\mathbf{0} = A^T r = A^T (b - Ax)$,

or $A^T A x = A^T b$

伪逆(Pseudo-inverse)和条件数

Nonsquare $m \times n$ matrix A has no inverse in usual sense

If rank(A)= n, *pseudoinverse* defined by

$$A^{+} = (A^T A)^{-1} A^T,$$

怎么记忆？

and

$$\text{cond}(A) = \|A\|_2 \cdot \|A^{+}\|_2$$

扩展条件数的概念到列满秩的非方阵

By conversion, cond(A) = ∞ if rank(A) < n

Just as condition number of square matrix measures closeness to singularity, condition number of rectangular matrix measures closeness to rank deficiency 非满秩，秩亏损

矩阵范数的定义并不限于方阵

Least squares solution of $Ax \cong b$ is given by

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

Sensitivity of full-rank linear least squares problem perturbing *b* only

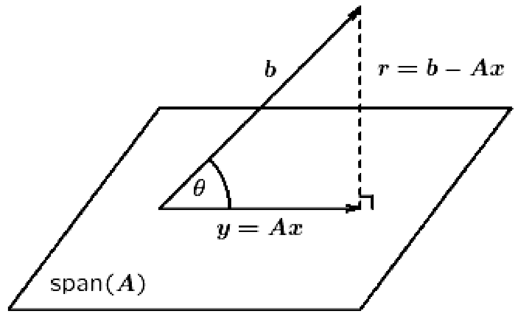
System of normal equations: $A^T Ax = A^T b$

Perturb b by Δb: $\quad A^T A(x+\Delta x) = A^T(b+\Delta b)$

It follows that $\quad A^T A\Delta x = A^T \Delta b$

and thus $\quad \Delta x = \left(A^T A\right)^{-1} A^T \Delta b = A^+ \Delta b$

Take norm: $\|\Delta x\|_2 = \|A^+ \Delta b\|_2 \le \|A^+\|_2 \|\Delta b\|_2$



Divide by $\|x\|_2$: $\quad \dfrac{\|\Delta x\|_2}{\|x\|_2} \le \|A^+\|_2 \dfrac{\|\Delta b\|_2}{\|x\|_2} = \text{cond}_2(A)\dfrac{\|\Delta b\|_2}{\|A\|_2\|x\|_2}$

$$\le \text{cond}_2(A)\frac{\|\Delta b\|_2}{\|Ax\|_2} = \text{cond}_2(A)\frac{\|b\|_2\|\Delta b\|_2}{\|Ax\|_2\|b\|_2}$$

θ为***b***和***Ax***的夹角

b与***Ax***接近程度影响敏感程度

So, $\quad \dfrac{\|\Delta x\|_2}{\|x\|_2} \le \text{cond}(A)\dfrac{1}{\cos(\theta)}\dfrac{\|\Delta b\|_2}{\|b\|_2}$

Sensitivity of linear least squares problem perturbing *A* only

System of normal equations: $A^T Ax = A^T b$

Perturb A by ΔA: $(A+\Delta A)^T(A+\Delta A)(x+\Delta x) = (A+\Delta A)^T b$

$$\text{L.H.S.} = A^T Ax + \Delta A^T Ax + A^T \Delta Ax + \Delta A^T \Delta Ax$$
$$+ A^T A\Delta x + \Delta A^T A\Delta x + A^T \Delta A\Delta x + \Delta A^T \Delta A\Delta x$$

Dropping small terms: $A^T Ax + \Delta A^T Ax + A^T \Delta Ax + A^T A\Delta x \approx A^T b + \Delta A^T b$

and thus $\Delta x \approx \left(A^T A\right)^{-1}\left(\underbrace{\Delta A^T b - \Delta A^T Ax}_{\text{factorize}} - A^T \Delta Ax\right) = \left(A^T A\right)^{-1}\left(\Delta A^T r - A^T \Delta Ax\right)$

Take norm: $\|\Delta x\|_2 \lesssim \|A^+\|_2 \|\Delta A\|_2 \|x\|_2 + \left\|\left(A^T A\right)^{-1}\right\|_2 \|\Delta A^T\|_2 \|r\|_2$

Divide by $\|x\|_2$: $\dfrac{\|\Delta x\|_2}{\|x\|_2} \le \text{cond}_2(A)\dfrac{\|\Delta A\|_2}{\|A\|_2} + \underbrace{\|A\|_2^2 \left\|\left(A^T A\right)^{-1}\right\|}_{\text{cond}_2(A)^2} \dfrac{\|\Delta A^T\| \|r\|}{\|A\|_2 \|A\|_2 \|x\|}$

$$\le \text{cond}_2(A)\frac{\|\Delta A\|_2}{\|A\|_2} + \text{cond}_2(A)^2 \frac{\|\Delta A^T\| \|r\|}{\|A\|_2 \|Ax\|_2}$$

若*b*与*Ax*接近，θ接近0；否则很病态

So, $\dfrac{\|\Delta x\|_2}{\|x\|_2} \lessapprox \left([\text{cond}(A)]^2 \tan(\theta) + \text{cond}(A)\right)\dfrac{\|\Delta A\|_2}{\|A\|_2}$

法方程方法的缺点

For full rank matrix *A*, the symmetric n×n matrix A^TA is positive definite, having Cholesky factorization:

$$A^TA = LL^T$$

Information can be lost in forming A^TA and A^Tb

由于浮点运算误差

For example, take

$$A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix},$$

where ε is positive number smaller than $\sqrt{\varepsilon_{\text{mach}}}$ **~10^{-8}**

Then in floating-point arithmetic

Normal equations method involves:

$$A^TA = \begin{bmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

which is singular

Rectangular matrix

↓

Square matrix

↓

triangular matrix

求解线性方程组问题的条件数

$$\text{cond}_2(A^TA) = [\text{cond}_2(A)]^2$$

后面将奇异值分解时证明

无论fit情况怎样，都有 condition-squaring effect, 所以算法不够稳定

Augmented System Method

Definition of residual and orthogonality requirement give $(m+n)\times(m+n)$ augmented system

$$\begin{bmatrix} I & A \\ A^T & O \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix}$$

System not positive definite, larger than original, and requires storing two copies of A

But allows greater freedom in choosing pivots in computing LDL^T or LU factorization

选主元很关键！
否则回到法方程法

Introducing scaling parameter α gives system

$$\begin{bmatrix} \alpha I & A \\ A^T & O \end{bmatrix} \begin{bmatrix} r/\alpha \\ x \end{bmatrix} = \begin{bmatrix} b \\ \mathbf{0} \end{bmatrix},$$

Reasonable rule of thumb

$$\alpha = \max_{i,j} |a_{ij}| / 1000$$

which allows control over relative weights of two subsystems in choosing pivots

Augmented system sometimes useful, but far from ideal in work and storage required

MATLAB中处理大型、稀疏问题?

利用QR正交化过程的解法

$$\boldsymbol{Ax} \cong \boldsymbol{b}$$

求$\boldsymbol{x}$, 使得 $\left\|\boldsymbol{b}-\boldsymbol{Ax}\right\|_2$ 达到最小.

$$\boldsymbol{A}=\boldsymbol{QR} \quad \Rightarrow \quad \boldsymbol{b}-\boldsymbol{Ax}=\boldsymbol{Q}(\boldsymbol{Q}^T\boldsymbol{b}-\boldsymbol{Rx})$$

由于$\boldsymbol{Q}$为正交阵，问题转化为求$\boldsymbol{x}$, 使得$\left\|\boldsymbol{Q}^T\boldsymbol{b}-\boldsymbol{Rx}\right\|_2$达到最小.

$$\boldsymbol{Q}^T\boldsymbol{b}-\boldsymbol{Rx}=\begin{bmatrix}\boldsymbol{Q}_1^T\\ \boldsymbol{Q}_2^T\end{bmatrix}\boldsymbol{b}-\begin{bmatrix}\boldsymbol{R}_1\\ \boldsymbol{O}\end{bmatrix}\boldsymbol{x}=\begin{bmatrix}\boldsymbol{Q}_1^T\boldsymbol{b}-\boldsymbol{R}_1\boldsymbol{x}\\ \boldsymbol{Q}_2^T\boldsymbol{b}\end{bmatrix}$$

$\boldsymbol{Q}_1$为$\boldsymbol{Q}$的前n列，$\boldsymbol{R}_1$为$n\times n$上三角阵，所以

$$\left\|\boldsymbol{Q}^T\boldsymbol{b}-\boldsymbol{Rx}\right\|_2^2=\left\|\boldsymbol{Q}_1^T\boldsymbol{b}-\boldsymbol{R}_1\boldsymbol{x}\right\|_2^2+\left\|\boldsymbol{Q}_2^T\boldsymbol{b}\right\|_2^2\geq\left\|\boldsymbol{Q}_2^T\boldsymbol{b}\right\|_2^2$$

并且当 $\boldsymbol{x}=\boldsymbol{R}_1^{-1}\boldsymbol{Q}_1^T\boldsymbol{b}$ 时取到最小值.

实际计算中，$\boldsymbol{Q}_1^T\boldsymbol{b}$ 是$\boldsymbol{Q}^T\boldsymbol{b}$ 前n个分量组成的向量，在将$\boldsymbol{A}$正交化为上三角矩阵的同时得到.

Householder QR Factorization

$$H_n \cdots H_1 A = \begin{bmatrix} R \\ O \end{bmatrix}$$

H_i是对称的正交阵

with R $n \times n$ and upper triangular

If $Q = H_1 \cdots H_n$, then

$$A = Q \begin{bmatrix} R \\ O \end{bmatrix}$$

To preserve solution of linear least squares problem, right-hand-side b transformed by same sequence of Householder transformations

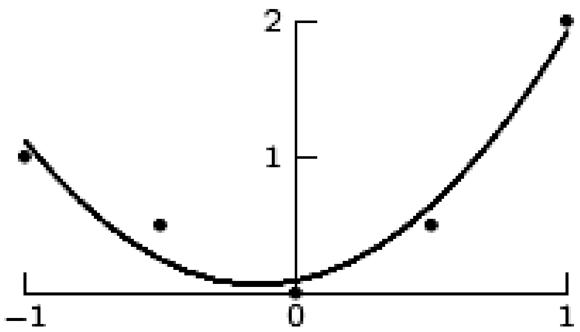
同时将这些Householder变换作用到b上

Then solve triangular least squares problem

$$\begin{bmatrix} R \\ O \end{bmatrix} x \cong Q^T b$$

for solution x of original least squares problem

Example: Householder QR Factorization



For polynomial data-fitting example given previously, with

$$A = \begin{bmatrix} 1 & -1.0 & 1.0 \\ 1 & -0.5 & 0.25 \\ 1 & 0.0 & 0.0 \\ 1 & 0.5 & 0.25 \\ 1 & 1.0 & 1.0 \end{bmatrix}, \quad b = \begin{bmatrix} 1.0 \\ 0.5 \\ 0.0 \\ 0.5 \\ 2.0 \end{bmatrix},$$

Householder vector v_1 for annihilating subdiagonal entries of first column of A is

$$v_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -2.236 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3.236 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Example Continued

Applying resulting Householder transformation H_1 yields transformed matrix and right-hand side

$$H_1 A = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & -0.191 & -0.405 \\ 0 & 0.309 & -0.655 \\ 0 & 0.809 & -0.405 \\ 0 & 1.309 & 0.345 \end{bmatrix}$$

$$H_1 b = \begin{bmatrix} -1.789 \\ -0.362 \\ -0.862 \\ -0.362 \\ 1.138 \end{bmatrix}$$

Example Continued

Householder vector v_2 for annihilating subdiagonal entries of second column of $H_1 A$ is

$$v_2 = \begin{bmatrix} 0 \\ -0.191 \\ 0.309 \\ 0.809 \\ 1.309 \end{bmatrix} - \begin{bmatrix} 0 \\ 1.581 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1.772 \\ 0.309 \\ 0.809 \\ 1.309 \end{bmatrix}$$

Applying resulting Householder transformation H_2 yields

$$H_2 H_1 A = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & -0.725 \\ 0 & 0 & -0.589 \\ 0 & 0 & 0.047 \end{bmatrix} \qquad H_2 H_1 b = \begin{bmatrix} -1.789 \\ 0.632 \\ -1.035 \\ -0.816 \\ 0.404 \end{bmatrix}$$

Householder vector v_3 for annihilating subdiagonal entries of third column of H_2H_1A is

$$v_3 = \begin{bmatrix} 0 \\ 0 \\ -0.725 \\ -0.589 \\ 0.047 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0.935 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1.660 \\ -0.589 \\ 0.047 \end{bmatrix}$$

Applying resulting Householder transformation H_3 yields

$$H_3H_2H_1A = \begin{bmatrix} -2.236 & 0 & -1.118 \\ 0 & 1.581 & 0 \\ 0 & 0 & 0.935 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad H_3H_2H_1b = \begin{bmatrix} -1.789 \\ 0.632 \\ 1.336 \\ 0.026 \\ 0.337 \end{bmatrix}$$

Now solve upper triangular system $Rx = c_1$ by back-substitution to obtain $x = [0.086 \quad 0.400 \quad 1.429]^T$

- 秩亏损(Rank Deficiency)
  - 如果矩阵A的秩<n, QR分解仍然存在，但得到列不满秩的上三角阵R (简化QR分解中的矩阵R为奇异上三角阵)

    Householder变换QR算法不会中断;
    R矩阵对角线上可能出现0
  - 怎么解线性最小二乘?

$$Q^T A x = \begin{bmatrix} R \\ O \end{bmatrix} x \cong \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = Q^T b,$$

  - 最小二乘解不唯一 (基本解、最小范数解)
    - 通过“列主元”的QR分解，求得基本解
    - 通过奇异值分解(SVD)，可求最小范数解
  - 近似秩亏损是常见的实际情况，如何检测它?

    常由不合理的问题建模造成

近似秩亏损

Example: Near Rank Deficiency

Consider 3×2 matrix

$$A = \begin{bmatrix} 0.641 & 0.242 \\ 0.321 & 0.121 \\ 0.962 & 0.363 \end{bmatrix}$$

Computing QR factorization,

$$R = \begin{bmatrix} 1.1997 & 0.4527 \\ 0 & 0.0002 \end{bmatrix}$$

R extremely close to singular (exactly singular to 3-digit accuracy of problem statement)

后果1 If R used to solve linear least squares problem, result is highly sensitive to perturbations in right-hand side

后果2 For practical purposes, $\text{rank}(A) = 1$ rather than 2, because columns nearly linearly dependent

QR with Column Pivoting

Instead of processing columns in natural order, select for reduction at each stage column of remaining unreduced submatrix having maximum Euclidean norm

列交换：
剩下的列中取模最大的

$$H_k \cdots H_1 A P_1 \cdots P_k = \begin{bmatrix} R & S \\ O & O \end{bmatrix}$$

秩同A的秩

If $\operatorname{rank}(A) = k < n$, then after k steps, norms of remaining unreduced columns will be zero (or "negligible" in finite-precision arithmetic) below row k

$$Q^T A P = \begin{bmatrix} R & S \\ O & O \end{bmatrix}$$

$$Q^T A x \cong Q^T b = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

$$\begin{bmatrix} R & S \\ O & O \end{bmatrix} P^T x \cong \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

设 $y = P^T x$，且 $y = \begin{bmatrix} z \\ o \end{bmatrix}$

$$Rz = c_1, \; x = P \begin{bmatrix} z \\ o \end{bmatrix}$$

- 求基本解
- 用一个阈值判断剩下的列是否为零，从而决定A的秩
- 用于求解 $Ax \cong b$ ，m<n的情况

Matlab有关命令的说明

- **qr命令**
  - 能处理稠密或“稀疏”矩阵, 允许**m<n (**列不满秩**)**
  - [Q,R] = qr(A); 完整的**QR**分解**, Householder, Givens**
  - [Q,R] = qr(A,0); 简化形式的**QR**分解，若**m<n**同上
  - [Q,R,E] = qr(A); **E**为排列阵，**AE=QR**，**R**对角元递减
  - [Q,R,E] = qr(A,0); **E**为排列向量，**A(:, E)=QR**
  - R = qr(A); **A**稠密时，返回矩阵上三角部分为**R; A**稀疏时，上三角阵**R**是上三角阵，且$R^TR=A^TA$
  - [C,R] = qr(A,B); 对稀疏的**A**，$C=Q^TB$
- **\命令**
  - 求解线性最小二乘问题，列重排的**Householder**变换
  - 列不满秩会报**warning**，得到一个基本解
- **polyfit命令**
  - 多项式最佳平方逼近：p = polyfit(x,y,n)

- 小结
  - 线性最小二乘问题
  - 问题的敏感性：依赖于b和span(A)的接近程度
  - 解法一：法方程法
  - 解法二：增广方程系统
  - 解法三：正交变换法（QR分解）
  - 列不满秩的情况：列选主元的QR分解
    - 确定矩阵A的秩
    - 对不满秩问题给出一个稳定解法

方法之间的比较后面讨论

正交变换方法数值稳定性很好！

Assignment

- 阅读课本第三章有关内容
- 阅读文献： **(“教学资源”--” top 10 algorithm”)**
- G. W. Stewart, “The decompositional approach to matrix computation,” Computing in Science and Engineering, Vol. 2, No. 1, pp. 50-59, 2000
- 作业题见网络学堂