清华大学统计学辅修课程

Computational Statistics

1

Lecture 4-
Linear Algebra

周在莹
清华大学统计学研究中心
http://www.stat.tsinghua.edu.cn

清华大学统计学研究中心

1

Outline

2

- Orthogonality
  - Projection
  - QR Factorization
  - Gram-Schmidt Orthogonalization
  - Householder Triangularization
  - Gaussian Elimination
- Decomposition
  - Cholesky Factorization/Decomposition
  - Eigenvalue Decomposition
  - Singular Value Decomposition (SVD)
  - Least Squares Problems

清华大学统计学研究中心

2

3

Projection

清华大学统计学研究中心

3

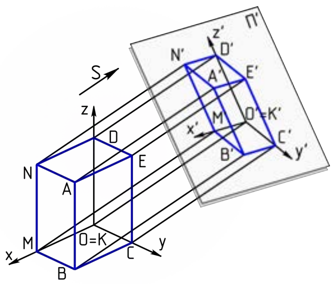
Projectors

4

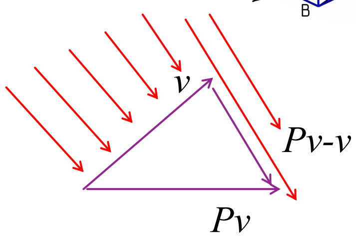
- A projector is an idempotent matrix, i.e. a square matrix P that satisfies

$$P^2 = P$$

- Physical picture: oblique projector, shine a light onto the subspace range(P) from just the direction Pv-v, then Pv would be the shadow projected by the vector v
- Pv-$v \in$ null(P)



- Whether or not all the projection lines are orthogonal to the projection plane



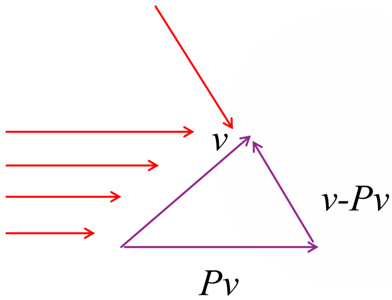
清华大学统计学研究中心

4

Complementary Projectors

5

- If P is a projector, $I - P$ is also a projector, called the complementary projector to P
- $I - P$ project onto range($I - P$)
- range($I - P$) = null(P)
- range(P) = null($I - P$)
- null($I - P$) ∩ null(P) = {0}
- range(P) ∩ null(P) = {0}
- P is the projector onto range (P) along range ($I - P$)
- Decomposition of a given vector: Using a projector we can decompose any vector v into $v = Pv + (I - P)v$. This decomposition is unique (Why?)



清华大学统计学研究中心

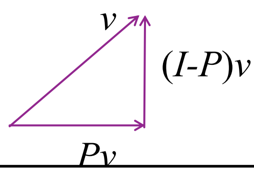
5

Orthogonal Projectors

6

- A projector P is orthogonal if range(P) and range($I - P$) are orthogonal (NOT orthogonal matrices!)
- A projector P is orthogonal if and only if $P = P^*$
- In a statistical context, P is a symmetric matrix
- Then $P = QQ^T$, where the columns of $Q_{m \times n}$ are orthonormal
- Denote $Q = [q_1 | q_2| \dots |q_n]$, then

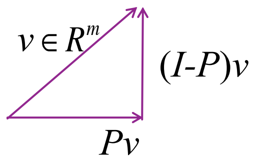
$$P = \sum_{j=1}^{n} q_j q_j^T, \quad q_j \in R^m$$



清华大学统计学研究中心

6

7

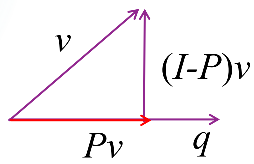


$$v = Pv + (I\text{-}P)v$$

$$P = \sum_{j=1}^{n} q_j q_j^T, \quad q_j \in R^m$$

- Rank-one orthogonal projector $P = qq^T$ isolates the component Pv in a single direction q. range(qq^T) = range(q)
- The complement $P = I - qq^T$ is a rank $m-1$ orthogonal projector that eliminates the component Pv in the direction of q
- q is a unit vector. For arbitrary nonzero vectors a:

$$\begin{cases} P = qq^T \\ P_\perp = I - qq^T \end{cases}$$ the projection matrix onto q



$$\begin{cases} P = \dfrac{aa^T}{a^T a} \\ P_\perp = I - \dfrac{aa^T}{a^T a} \end{cases}$$

清华大学统计学研究中心

7

Projection with an Arbitrary Basis

8

- Suppose that the subspace is spanned by the linearly independent vectors $\{ a_1 , \ldots , a_n\}$
- Let A be the $m \times n$ matrix whose jth column is a_j
- For v, consider its orthogonal projection onto range(A), denoted as y
- $v\text{-}y$ must be orthogonal to range(A)

$$\boxed{\begin{array}{l} a_j^T(v\text{-}y) = 0 \\ \text{for every } j \end{array}} \xRightarrow{y=Ax} \boxed{\begin{array}{l} a_j^T(v\text{-}Ax) = 0 \\ \text{for every } j \end{array}} \Rightarrow \boxed{A^T(v\text{-}Ax) = 0}$$

$$\Rightarrow \boxed{x = (A^TA)^{-1}A^Tv} \xRightarrow{y=Ax} \boxed{P = A\,(A^TA)^{-1}A^T} \qquad P = \frac{aa^T}{a^T a}$$

清华大学统计学研究中心

8

QR Factorization

9

清华大学统计学研究中心

9

Reduced QR Factorization

10

- For many applications, we find ourselves interested in the column spaces of a matrix A

$$\langle a_1 \rangle \subseteq \langle a_1, a_2 \rangle \subseteq \langle a_1, a_2, a_3 \rangle \subseteq \ldots$$

- The idea of QR factorization is the construction of a sequence of orthonormal vectors q_1, q_2, ... that span these successive spaces
- This amounts to

$$A = \hat{Q}\hat{R} \quad \left[a_1 | a_2 | \ldots | a_n \right] = \left[q_1 | q_2 | \ldots | q_n \right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}, r_{jj} \neq 0$$

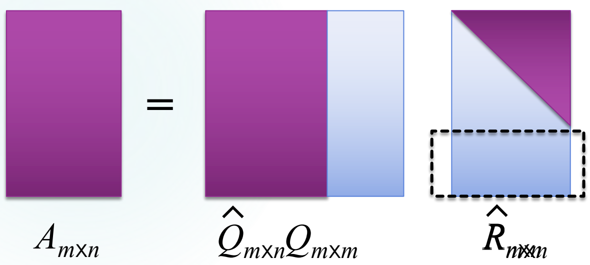
when A_{mxn} has full rank n

清华大学统计学研究中心

10

Full QR Factorization

- $Q_{mxn} = [q_1 | q_2 | \ldots]$, appending additional $m - n$ orthonormal columns to make it unitary
- So rows of zeros should be appended to R so that it becomes taller, still upper-triangular



The added columns span a space orthogonal to range(A), constitute an orthonormal basis for null(A^T)

11

Gram-Schmidt Orthogonalization

$$\left[\begin{array}{c|c|c|c} a_1 & a_2 & \cdots & a_n \end{array}\right] = \left[\begin{array}{c|c|c|c} q_1 & q_2 & \cdots & q_n \end{array}\right] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix},$$

$r_{jj} \neq 0$

$$q_1 = \frac{a_1}{r_{11}},$$
$$q_2 = \frac{a_2 - r_{12} q_1}{r_{22}},$$
$$q_3 = \frac{a_3 - r_{13} q_1 - r_{23} q_2}{r_{33}},$$
$$\vdots$$
$$q_n = \frac{a_n - \sum_{i=1}^{n-1} r_{in} q_i}{r_{nn}}.$$

- where

$$r_{ij} = q_i^T a_j \ (i \neq j),$$
$$r_{jj} = \left\| a_j - \sum_{i=1}^{j-1} r_{ij} q_i \right\|_2.$$

12

Algorithm of Classical Gram-Schmidt (Unstable)

13

for $j = 1$ to n
  $v_j = a_j$
  for $i = 1$ to $j - 1$
    $r_{ij} = q_i^T a_j$
    $v_j = v_j - r_{ij} q_i$
  $r_{jj} = ||v_j||_2$
  $q_j = v_j / r_{jj}$

- Mathematically, it offers a simple route to understanding and proving various properties of QR factorizations
- Numerically, it turns out to be unstable because of rounding errors on a computer.
- To emphasize the instability, numerical analysts refer to this as the **classical** Gram-Schmidt iteration, as opposed to the **modified** Gram-Schmidt iteration, discussed later

清华大学统计学研究中心

13

Existence and Uniqueness

14

- All matrices have QR factorizations, and under suitable restrictions, they are unique
- Theorem 1. Every $m \times n$ matrix A $(m \geq n)$ has a full QR factorization, hence also a reduced QR factorization
- Theorem 2. Each $m \times n$ matrix A $(m \geq n)$ of full rank has a unique **reduced** QR factorization $A = \hat{Q}\hat{R}$ with $r_{jj} > 0$

清华大学统计学研究中心

14

Solution of $Ax = b$ by QR Factorization

15

- Suppose we wish to solve $Ax = b$ for x, where m x m matrix A is nonsingular
- If $A = QR$ is a QR factorization, then we can write $QRx = b$, or $Rx = Q^T b$
- Method for computing the solution to $Ax = b$:
  1. Compute a QR factorization $A = QR$
  2. Compute $y = Q^T b$
  3. Solve $Rx = y$ for x
- However, Gaussian elimination is the algorithm generally used in practice, since it requires only half as many numerical operations

清华大学统计学研究中心

15

16

Gram-Schmidt Orthogonalization

清华大学统计学研究中心

16

Inner Products and Orthogonality

17

- The most important idea to draw from these concepts is: inner products can be used to decompose arbitrary vectors into orthogonal components
- For example, suppose that $\{q_1, q_2, \cdots, q_n\}$ is an orthonormal set, and let v be an arbitrary vector. Then

$$r = v - (q_1^* v)q_1 - (q_2^* v)q_2 - \cdots - (q_n^* v)q_n = (I - \sum q_i q_1^*)v$$

is orthogonal to $\{q_1, q_2, \cdots, q_n\}$.

Thus we see that v can be decomposed into $n + 1$ orthogonal components: $v = r + \sum q_i q_1^* v$

- Interpretations: View v as a sum of coefficients $q_i^* v$ times vectors q_i; View v as a sum of orthogonal projections of v onto the various directions q_i. The ith projection operation is achieved by the rank-one matrix $q_i q_1^*$

清华大学统计学研究中心

17

Gram-Schmidt Projections

18

- Use orthogonal projectors to describe the G-S iteration in another way
- Suppose that $m \times n$ matrix A ($m \geq n$) has full rank with columns $\{a_j\}$
- Let Q_{j-1} denote the $m \times (j-1)$ matrix containing the first $j-1$ columns of Q, $Q_{j-1} = [q_1 \mid \ldots \mid q_{j-1}]$
- $$P_j = I - Q_{j-1} Q_{j-1}^T$$

  is the $m \times m$ matrix of rank $m - (j-1)$ that projects R^m orthogonally onto the space orthogonal to $< q_1, \ldots, q_{j-1} >$, $j = 2, \ldots, n$, $P_1 = I$
- P_j is the $m \times m$ matrix of rank m - (j - 1) that projects R^m orthogonally onto the space orthogonal to $< q_1, \ldots, q_{j-1} >$, $j = 2, \ldots, n$, $P_1 = I$

$$q_1 = \frac{P_1 a_1}{\|P_1 a_1\|}, \quad q_2 = \frac{P_2 a_2}{\|P_2 a_2\|}, \ldots, \quad q_n = \frac{P_n a_n}{\|P_n a_n\|}$$

- q_j is orthogonal to $q_1, \ldots, q_{j-1}$, lies in the space $< a_1, \ldots, a_j >$, and has norm 1
- $v_j = P_j a_j = a_j - (q_1 q_1^T) a_j - \ldots - (q_{j-1} q_{j-1}^T) a_j = a_j - (q_1^T a_j) q_1 - \ldots - (q_{j-1}^T a_j) q_{j-1}$

```
for j = 1 to n
  v_j = a_j
  for i = 1 to j - 1
    r_ij = q_i^T a_j
    v_j = v_j - r_ij q_i
  r_jj = ||v_j||
  q_j = v_j / r_jj
```

清华大学统计学研究中心

18

Modified Gram-Schmidt Algorithm

19

- Compute the same result by a sequence of $j-1$ projections of rank $m-1$

$$P_j = P_{\perp q_{j-1}} \cdots P_{\perp q_2} P_{\perp q_1}$$

$$v_j = P_j a_j$$

$$v_j^{(1)} = a_j,$$

$$v_j^{(2)} = P_{\perp q_1} v_j^{(1)} = v_j^{(1)} - q_1 q_1^T v_j^{(1)},$$

$$v_j^{(3)} = P_{\perp q_2} v_j^{(2)} = v_j^{(2)} - q_2 q_2^T v_j^{(2)},$$

$$\vdots$$

$$v_j = v_j^{(j)} = P_{\perp q_{j-1}} v_j^{(j-1)} = v_j^{(j-1)} - q_{j-1} q_{j-1}^T v_j^{(j-1)}.$$

for $i = 1$ to n
  $v_i = a_i$
for $i = 1$ to n
  $r_{ii} = ||v_i||$
  $q_i = v_i / r_{ii}$
  for $j = i + 1$ to n
    $r_{ij} = q_i^T v_j$
    $v_j = v_j - r_{ij} q_i$

清华大学统计学研究中心

19

Comparison

20

Classical G-S

for $j = 1$ to n
  $v_j = a_j$
  for $i = 1$ to $j - 1$
    $r_{ij} = q_i^T a_j$
    $v_j = v_j - r_{ij} q_i$
  $r_{jj} = ||v_j||$
  $q_j = v_j / r_{jj}$

Modified G-S

for $i = 1$ to n
  $v_i = a_i$
for $i = 1$ to n
  $r_{ii} = ||v_i||$
  $q_i = v_i / r_{ii}$
  for $j = i + 1$ to n
    $r_{ij} = q_i^T v_j$
    $v_j = v_j - r_{ij} q_i$

C/M G-S

for $j = 1$ to n
  $v_j = a_j$
  for $i = 1$ to $j - 1$
    $r_{ij} = q_i^T a_j$ (CGS)
    $r_{ij} = q_i^T v_j$ (MGS)
    $v_j = v_j - r_{ij} q_i$
  $r_{jj} = ||v_j||$
  $q_j = v_j / r_{jj}$

- Modified G-S is numerically stable (less sensitive to rounding errors)

清华大学统计学研究中心

20

Example: Classical vs. Modified Gram-Schmidt

21

- Compare classical and modified G-S for the vectors
  $a_1 = (1, \varepsilon, 0, 0)^T$, $a_2 = (1, 0, \varepsilon, 0)^T$, $a_3 = (1, 0, 0, \varepsilon)^T$
  making the approximation $1 + \varepsilon^2 \approx 1$
- Classical:
  - $v_1 \leftarrow (1, \varepsilon, 0,0)^T, r_{11} = \sqrt{1+\varepsilon^2} \approx 1, q_1 = v_1/1 = (1, \varepsilon, 0,0)^T$
  - $v_2 \leftarrow (1,0, \varepsilon, 0)^T, r_{12} = q_1^T a_2 = 1, v_2 \leftarrow v_2 - 1q_1 = (0, -\varepsilon, \varepsilon, 0)^T$
    $r_{22} = \sqrt{2}\varepsilon, q_2 = v_2/r_{22} = (0,-1,1,0)^T/\sqrt{2}$
  - $v_3 \leftarrow (1,0,0, \varepsilon)^T, r_{13} = q_1^T a_3 = 1, v_3 \leftarrow v_3 - 1q_1 = (0, -\varepsilon, 0, \varepsilon)^T$
    $r_{23} = q_2^T a_3 = 0, v_3 \leftarrow v_3 - 0q_2 = (0, -\varepsilon, 0, \varepsilon)^T, r_{33} = \sqrt{2}\varepsilon,$
    $q_3 = v_3/r_{33} = (0,-1,0,1)^T/\sqrt{2}$

清华大学统计学研究中心

21

22

- Modified:
  - $v_1 \leftarrow (1, \varepsilon, 0,0)^T, r_{11} = \sqrt{1+\varepsilon^2} \approx 1, q_1 = v_1/1 = (1, \varepsilon, 0,0)^T$
  - $v_2 \leftarrow (1,0, \varepsilon, 0)^T, r_{12} = q_1^T v_2 = 1, v_2 \leftarrow v_2 - 1q_1 = (0, -\varepsilon, \varepsilon, 0)^T$
    $r_{22} = \sqrt{2}\varepsilon, q_2 = v_2/r_{22} = (0,-1,1,0)^T/\sqrt{2}$
  - $v_3 \leftarrow (1,0,0, \varepsilon)^T, r_{13} = q_1^T v_3 = 1, v_3 \leftarrow v_3 - 1q_1 = (0, -\varepsilon, 0, \varepsilon)^T$
    $r_{23} = q_2^T v_3 = \varepsilon/\sqrt{2}, v_3 \leftarrow v_3 - r_{23}q_2 = (0, -\varepsilon/2, -\varepsilon/2, \varepsilon)^T, r_{33} = \sqrt{6}\varepsilon/2,$
    $q_3 = v_3/r_{33} = (0,-1,-1,2)^T/\sqrt{6}$
- Check orthogonality:
  - Classical: $q_2^T q_3 = (0,-1,1,0)(0,-1,0,1)^T/2 = 1/2$
  - Modified: $q_2^T q_3 = (0,-1,1,0)(0,-1,-1,2)^T/\sqrt{12} = 0$

清华大学统计学研究中心

22

Operation Count

23

- Count number of floating points operations – "flops" – in an algorithm
- Each +, −, ×, /, or $\sqrt{\ }$ counts as one flop
- No distinction between real and complex
- No consideration of memory accesses or other performance aspects

清华大学统计学研究中心

23

Operation Count - Modified G-S

24

- Example: Count all +, −, x, / in the Modified Gram-Schmidt algorithm (not just the leading term)

(1)	**for** $i = 1$ **to** n	
(2)	$\quad v_i = a_i$	
(3)	**for** $i = 1$ **to** n	
(4)	$\quad r_{ii} = \|\|v_i\|\|$	m multiplications, $m - 1$ additions
(5)	$\quad q_i = v_i / r_{ii}$	m divisions
(6)	$\quad$ **for** $j = i + 1$ **to** n	
(7)	$\qquad r_{ij} = q_i^T v_j$	m multiplications, $m - 1$ additions
(8)	$\qquad v_j = v_j - r_{ij} q_i$	m multiplications, m subtractions

清华大学统计学研究中心

24

- The total for each operation is

$$\# A = \sum_{i=1}^{n}\left(m-1+\sum_{j=i+1}^{n} m-1 \right) = n(m-1)+\sum_{i=1}^{n}(m-1)(n-i)$$
$$= n(m-1)+\frac{n(n-1)(m-1)}{2} = \frac{1}{2}n(n+1)(m-1)$$
$$\# S = \sum_{i=1}^{n}\sum_{j=i+1}^{n} m = \sum_{i=1}^{n} m(n-i) = \frac{1}{2}mn(n-1)$$
$$\# M = \sum_{i=1}^{n}\left(m+\sum_{j=i+1}^{n} 2m \right) = mn+\sum_{i=1}^{n} 2m(n-i)$$
$$= mn+\frac{2mn(n-1)}{2} = mn^2$$
$$\# D = \sum_{i=1}^{n} m = mn$$

清华大学统计学研究中心

25

25

- And the total flop count is

$$\frac{1}{2}n(n+1)(m-1)+\frac{1}{2}mn(n-1)+mn^2+mn$$
$$= 2mn^2+mn-\frac{1}{2}n^2-\frac{1}{2}n \sim 2mn^2$$

- The symbol ~ indicates asymptotic value as $m, n \to \infty$ (leading term)
- Easier to find just the leading term:

– Most work done in lines (7) and (8), with $4m$ flops per iteration

– Including the loops, the total becomes

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n} 4m = 4m\sum_{i=1}^{n}(n-i) \sim 4m\sum_{i=1}^{n} i \sim 2mn^2$$

清华大学统计学研究中心

26

26

27

Householder Triangularization

清华大学统计学研究中心

27

28

Gram-Schmidt as Triangular Orthogonalization

- Gram-Schmidt multiplies with triangular matrices to make columns orthogonal, for example at the first step:

$$\left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right]\begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}} & \cdots & \frac{-r_{1n}}{r_{11}} \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} = \left[\begin{array}{c|c|c|c} q_1 & v_2^{(2)} & \cdots & v_n^{(2)} \end{array}\right]$$

- After all the steps we get a product of triangular matrices

$$A\underbrace{R_1R_2\ldots R_n}_{R^{-1}} = Q$$

- Triangular orthogonalization

清华大学统计学研究中心

28

Householder Triangularization

29

- The Householder method multiplies by unitary matrices to make columns triangular, for example at the first step of :

$$Q_1 A = \begin{bmatrix} r_{11} & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix}$$

- After all the steps we get a product of orthogonal matrices matrices

$$\underbrace{Q_n \dots Q_2 Q_1}_{Q^{-1}} A = R$$

- Orthogonal triangularization

- Numerically more stable than Gram-Schmidt orthogonalization.
- Principal method for computing QR factorizations.

清华大学统计学研究中心

29

Introducing Zeros

30

- Q_k introduces zeros below the diagonal in column k
- Preserves all the zeros previously introduced

$$\underset{A}{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}} \xrightarrow{Q_1} \underset{Q_1 A}{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}} \xrightarrow{Q_2} \underset{Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}} \xrightarrow{Q_3} \underset{Q_3 Q_2 Q_1 A}{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}$$

清华大学统计学研究中心

30

Householder Reflectors

- Let Q_k be of the form

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$$

where I is $(k-1) \times (k-1)$ and F is $(m-k+1) \times (m-k+1)$

- Create Householder reflector F that introduces zeros:

$$x = \begin{bmatrix} \times \\ \times \\ \vdots \\ \times \end{bmatrix} \in R^{m-k+1} \xrightarrow{F} Fx = \begin{bmatrix} \|x\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \|x\| e_1$$

- Idea: Reflect across hyperplane H orthogonal to $v = \|x\|e_1 - x$, by the unitary matrix

unitary/orthogonal, full rank$(m-k+1)$

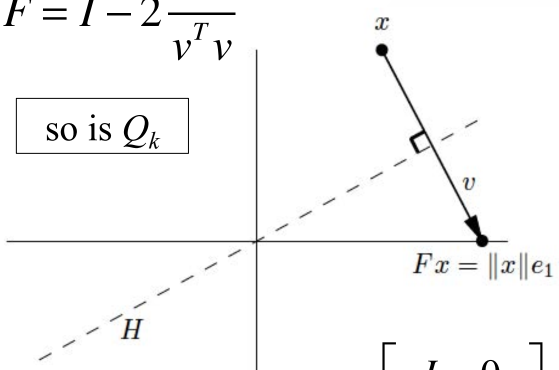
$$F = I - 2\frac{vv^T}{v^T v}$$

hermitian/symmetic

- Compare with projector

so is Q_k

$$P_{\perp v} = I - \frac{vv^T}{v^T v}$$



rank $m-1$

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}$$

Choice of Reflector

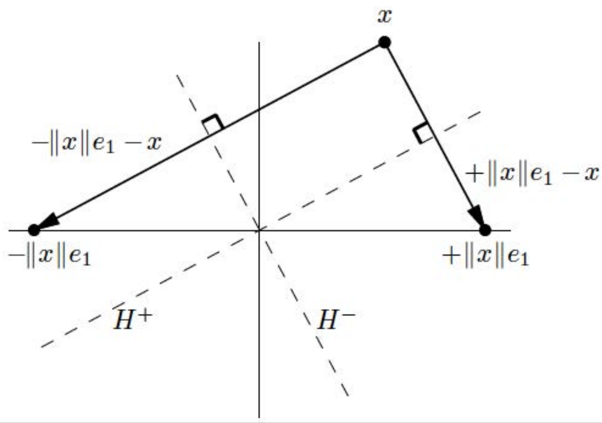
33

- The vector x can be reflected to any multiple z of $||x||e_1$ with $|z| = 1$
- Better numerical properties with large $||v||$, for example

$v = \text{sign}(x_1)\ ||x||e_1 + x$

- Note:

Here sign(0) =1



清华大学统计学研究中心

33

The Householder Algorithm

34

- Compute the factor R of a QR factorization of a given $m \times n$ matrix A ($m \geq n$)
- Leave result in place of A, store reflection vectors v_k for later use

<u>Householder QR Factorization</u>

for $k = 1$ to n

$x = A_{k:m,k}$

$v_k = sign(x_1)||x||e_1 + x$

$v_k = v_k/||v_k||$

$A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^T A_{k:m,k:n})$

清华大学统计学研究中心

34

Applying or Forming Q

35

- In solving $Ax = b$, compute $Q^{-1}b = Q_n \cdots Q_2Q_1b$ implicitly

Implicit Calculation of $Q^{-1}b$
for $k = 1$ to n
$b_{k:m} = b_{k:m} - 2v_k(v_k^T b_{k:m})$

- Compute $Qx = Q_1Q_2 \cdots Q_nx$ implicitly

Implicit Calculation of Qx
for $k = n$ down to 1
$x_{k:m} = x_{k:m} - 2v_k(v_k^T x_{k:m})$

- To create Q explicitly, apply to $x = e_1,\ldots e_m$
(or just $x = e_1,\ldots e_n$ for reduced Q)

清华大学统计学研究中心

35

Operation Count - Householder QR

36

- Most work is done by

$$A_{k:m,k:n} = A_{k:m,k:n} - 2v_k(v_k^T A_{k:m,k:n})$$

- Operations per iteration:
  - $2(m-k)(n-k)$ for the dot products $v_k^T A_{k:m,k:n}$
  - $(m-k)(n-k)$ for the outer product $2v_k(\ldots)$
  - $(m-k)(n-k)$ for the subtraction $A_{k:m,k:n} - \ldots$
  - $4(m-k)(n-k)$ total
- Including the outer loop, the total becomes

$$\sum_{k=1}^{n} 4(m-k)(n-k) = 4\sum_{k=1}^{n}\left(mn - k(m+n) + k^2\right)$$

$$\sim 4mn^2 - 4(m+n)n^2/2 + 4n^3/3 = 2mn^2 - 2n^3/3$$

清华大学统计学研究中心

36

Givens Rotations

37

- Alternative to Householder reflectors
- A Givens rotation

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

rotates $x \in R^2$ by θ

- To set an element to zero, choose $\cos\theta$ and $\sin\theta$ so that

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} \sqrt{x_i^2 + x_j^2} \\ 0 \end{bmatrix}$$

or

$$\cos\theta = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad \sin\theta = \frac{-x_j}{\sqrt{x_i^2 + x_j^2}}$$

清华大学统计学研究中心

37

Givens QR

38

- Introduce zeros in column from bottom and up

$$\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \end{bmatrix} \xrightarrow{(2,3)} \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ 0 & \times & \times \\ & \times & \times \end{bmatrix} \xrightarrow{(1,2)}$$

$$\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ & \times & \times \\ & \times & \times \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & \times & \times \\ & 0 & \times \end{bmatrix} \xrightarrow{(2,3)} \begin{bmatrix} \times & \times & \times \\ & \times & \times \\ & 0 & \times \\ & & \times \end{bmatrix} \xrightarrow{(3,4)} R$$

- Flop count $3mn^2 - n^3$ (50% more than Householder QR)

清华大学统计学研究中心

38

Gaussian Elimination

清华大学统计学研究中心

39

39

LU Factorization

40

- Gaussian elimination transforms a full linear system into an upper-triangular one by applying simple linear transformations on the left
- For $A_{m \times m}$, $\underbrace{L_{m-1} \dots L_2 L_1}_{L^{-1}} A = U$,

$$\boxed{A = LU}$$

$$\underset{A}{\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix}} \xrightarrow{L_1} \underset{L_1 A}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}} \xrightarrow{L_2} \underset{L_2 L_1 A}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & 0 & \times & \times \\ & 0 & \times & \times \end{bmatrix}} \xrightarrow{L_3} \underset{L_3 L_2 L_1 A}{\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & 0 & \times \end{bmatrix}}$$

- Triangular triangularization

清华大学统计学研究中心

40

Two Strokes of Luck

41

- Example

$$A=\begin{bmatrix}2&1&1&0\\4&3&3&1\\8&7&9&5\\6&7&9&8\end{bmatrix}$$

$$L_1A=\begin{bmatrix}1&&&\\-2&1&&\\-4&&1&\\-3&&&1\end{bmatrix}\begin{bmatrix}2&1&1&0\\4&3&3&1\\8&7&9&5\\6&7&9&8\end{bmatrix}=\begin{bmatrix}2&1&1&0\\&1&1&1\\&3&5&5\\&4&6&8\end{bmatrix}$$

$$L_2L_1A=\begin{bmatrix}1&&&\\&1&&\\&-3&1&\\&-4&&1\end{bmatrix}\begin{bmatrix}2&1&1&0\\&1&1&1\\&3&5&5\\&4&6&8\end{bmatrix}=\begin{bmatrix}2&1&1&0\\&1&1&1\\&&2&2\\&&2&4\end{bmatrix}$$

清华大学统计学研究中心

41

42

$$L_3L_2L_1A=\begin{bmatrix}1&&&\\&1&&\\&&1&\\&&-1&1\end{bmatrix}\begin{bmatrix}2&1&1&0\\&1&1&1\\&&2&2\\&&2&4\end{bmatrix}=\begin{bmatrix}2&1&1&0\\&1&1&1\\&&2&2\\&&&2\end{bmatrix}=U$$

- L_k^{-1} is obtained by negating its subdiagonal entries

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

- $L=L_1^{-1}L_2^{-1}L_3^{-1}$ is just the unit lower-triangular matrix with their nonzero subdiagonal entries inserted in the appropriate places

$$\underset{A}{\begin{bmatrix}2&1&1&0\\4&3&3&1\\8&7&9&5\\6&7&9&8\end{bmatrix}}=\underset{L}{\begin{bmatrix}1&&&\\2&1&&\\4&3&1&\\3&4&1&1\end{bmatrix}}\underset{U}{\begin{bmatrix}2&1&1&0\\&1&1&1\\&&2&2\\&&&2\end{bmatrix}}$$

清华大学统计学研究中心

42

General Formulas

43

- Suppose x_k denotes the kth column of the matrix at the beginning of step k

$$x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ x_{k+1,k} \\ \vdots \\ x_{mk} \end{bmatrix} \xrightarrow{L_k} L_k x_k = \begin{bmatrix} x_{1k} \\ \vdots \\ x_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- To do this, we subtract l_{jk} times row k from row j, where l_{jk} is the multiplier

$$l_{jk} = \frac{x_{jk}}{x_{kk}} \quad (k < j \le m)$$

- The matrix L_k takes the form

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{mk} & & & 1 \end{bmatrix}$$

清华大学统计学研究中心

43

Explain the Good Fortune

44

- Define $l_k = [0,\dots, 0, l_{k+1,k}, \dots l_{m,k}]^T$, then $L_k = I - l_k e_k^T$
  - The inverse of L_k is $I + l_k e_k^T$

    $(I - l_k e_k^T)(I + l_k e_k^T) = I - l_k e_k^T l_k e_k^T = I$

  - For

$$L = L_1^{-1} L_2^{-1} \cdots L_{m-1}^{-1} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{m1} & l_{m2} & \cdots & l_{m,m-1} & 1 \end{bmatrix}$$

$$L_k^{-1} L_{k+1}^{-1} = (I + l_k e_k^T)(I + l_{k+1} e_{k+1}^T) = I + l_k e_k^T + l_{k+1} e_{k+1}^T$$

清华大学统计学研究中心

44

Algorithm

45

Gaussian Elimination

$U = A,\ L = I$
for $k = 1$ to $m - 1$
 for $j = k + 1$ to m
  $l_{jk} = u_{jk}/u_{kk}$
  $u_{j,k:m} = u_{j,k:m} - l_{jk}u_{k,k:m}$

- Operation count

$u_{j,k:m} = u_{j,k:m} - l_{jk}u_{k,k:m}$

$$\sum_{k=1}^{m-1} 2(m-k+1)(m-k) \sim \frac{2}{3}m^3 \ flops$$

清华大学统计学研究中心

45

Solution of $Ax = b$ by LU Factorization

46

- $Ax = b$ is reduced to $LUx = b$.
- Solve two triangular systems:
  - first $Ly = b$ for the unknown y (forward substitution)
  - then $Ux = y$ for the unknown x (back substitution)
- Operation count $\boxed{l_{k1}y_1 + \ldots + l_{kk}y_k = b_k}$

$$\sum_{k=1}^{m} 2k \sim m^2 \ flops$$

- So if Gaussian elimination is used, the total work is ~ $2/3m^3$ flops, half the figure of ~$4/3m^3$ flops for a solution by Householder triangularization
- Usually used rather than QR factorization to solve square systems of equations

清华大学统计学研究中心

46

Instability of Gaussian Elimination

47

- Consider $$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

This matrix has full rank and is well-conditioned. Nevertheless, Gaussian elimination fails at the first step

- A slight perturbation of the same matrix reveals the more general problem.

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix} \quad \tilde{L}\tilde{U} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix} U = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1-10^{20} \end{bmatrix} \tilde{U} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

清华大学统计学研究中心

47

Pivoting

48

- The instability can be controlled by <u>permuting the order of the rows</u> of the matrix being operated on, an operation called pivoting
- Pivoting has been a standard feature of Gaussian elimination computations since the 1950s

$$\text{pivot} \quad \begin{bmatrix} \times & \times & \times & \times & \times \\ & \boxed{x_{kk}} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{kk} & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}$$

清华大学统计学研究中心

48

49

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & x_{ij} & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & 0 & \times & \times \\ & \times & 0 & \times & \times \\ & \times & x_{ij} & \times & \times \\ & \times & 0 & \times & \times \end{bmatrix}$$

- Free to choose any nonzero entry of $X_{k:m,k:m}$ as the pivot
- In practice, it is common to pick the largest possible
- At step k, rows/columns of the working matrix shall be permuted so as to move the pivot into the (k, k) position

50

Partial Pivoting

- Only rows are interchanged

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{bmatrix} \xrightarrow{L_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}$$

Pivot selection　　Row interchange　　Elimination

- $L_{m-1}P_{m-1} \ldots L_2P_2L_1P_1A = U$

$PA = LU$ Factorization and a Third Stroke of Luck

51

- $L_{m\text{-}1}P_{m\text{-}1}\ldots L_2P_2L_1P_1A = U$

$L_{m\text{-}1}P_{m\text{-}1}\ldots L_2P_2L_1P_1 = (L_{m\text{-}1}'\ldots L_2'L_1')(P_{m\text{-}1}\ldots P_2P_1) = LP$

- For example,

$$L_3P_3L_2P_2L_1P_1 = L_3'L_2'L_1'P_3P_2P_1$$

with $L_3' = L_3$, $L_2' = P_3L_2P_3^{-1}$, $L_1' = P_3P_2L_1P_2^{-1}P_3^{-1}$

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{mk} & & & 1 \end{bmatrix} \xrightarrow{P_{k+1,m}L_k} \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{mk} & & & \\ & & \vdots & & \ddots & \\ & & -l_{k+1,k} & 1 & & 0 \end{bmatrix} \xrightarrow{P_{k+1,m}L_kP_{k+1,m}^{-1}} \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{mk} & 1 & \\ & & \vdots & & \ddots \\ & & -l_{k+1,k} & & & 1 \end{bmatrix}$$

$$L_k' = P_{m\text{-}1}\ldots P_{k+1}L_kP_{k+1}^{-1}\ldots P_{m\text{-}1}^{-1}$$

清华大学统计学研究中心

51

Algorithm

52

- $PA = LU$
-

Gaussian Elimination with Partial Pivoting

$U = A,\ L = I,\ P = I$

for $k = 1$ to $m - 1$

    Select $i \geq k$ to maximize $|u_{ik}|$

    $u_{k,k:m} \leftrightarrow u_{i,k:m}$ (interchange two rows)

    $l_{k,1:k\text{-}1} \leftrightarrow l_{i,1:k\text{-}1}$

    $P_{k,:} \leftrightarrow P_{i,:}$

    for $j = k + 1$ to m

        $l_{jk} = u_{jk}/u_{kk}$

        $u_{j,k:m} = u_{j,k:m} - l_{jk}u_{k,k:m}$

清华大学统计学研究中心

52

Complete Pivoting

53

- Each elimination step with a permutation P_k of the rows applied on the left, a permutation Q_k of the columns applied on the right:

$$L_{m-1}P_{m-1} \ldots L_2P_2L_1P_1\, A\, Q_1Q_2 \ldots Q_{m-1} = U$$

$$(L_{m-1}' \ldots L_2'L_1')(P_{m-1} \ldots P_2P_1)\, A\, (Q_1Q_2 \ldots Q_{m-1}) = U$$

- $\underline{PAQ = LU}$
- The selection of pivots takes a significant amount of time
- In practice this is rarely done, because the improvement in stability is marginal

清华大学统计学研究中心

53

Fundamentals: Matrix-Vector Multiplication

54

清华大学统计学研究中心

54

A Matrix Times a Vector

55

$$b = Ax = \sum_{j=1}^{n} x_j a_j$$

$$\begin{bmatrix} \\ b \\ \\ \end{bmatrix} = \begin{bmatrix} & & & \\ a_1 & a_2 & \cdots & a_n \\ & & & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} \\ a_1 \\ \\ \end{bmatrix} + x_2 \begin{bmatrix} \\ a_2 \\ \\ \end{bmatrix} + \cdots x_n \begin{bmatrix} \\ a_n \\ \\ \end{bmatrix}$$

- b is expressed as a linear combination of the columns a_j

清华大学统计学研究中心

55

Example: Vandermonde Matrix

56

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$$

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{n-1} \end{bmatrix} \quad c = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix}$$

$(Ac)_i = p(x_i)$ gives the sampled polynomial values

- A is a linear map from vectors of coefficients of polynomials p of degree $< n$ to vectors $(p(x_1), p(x_2), \ldots, p(x_m))$ of sampled polynomial values

$$A = \begin{bmatrix} & & & & \\ 1 & x & x^2 & \cdots & x^{n-1} \\ & & & & \end{bmatrix}$$

$Ac = p(x)$, a single vector summation that at once gives a linear combination of these monomials

清华大学统计学研究中心

56

A Matrix Times a Matrix

57

$$B = AC, \quad b_j = Ac_j = \sum_{k=1}^{m} c_{kj} a_k$$

$$\begin{bmatrix} \Bigg| & \Bigg| & & \Bigg| \\ b_1 & b_2 & \cdots & b_n \\ \Bigg| & \Bigg| & & \Bigg| \end{bmatrix} = \begin{bmatrix} \Bigg| & \Bigg| & & \Bigg| \\ a_1 & a_2 & \cdots & a_m \\ \Bigg| & \Bigg| & & \Bigg| \end{bmatrix} \begin{bmatrix} \Big| & \Big| & & \Big| \\ c_1 & c_2 & \cdots & c_n \\ \Big| & \Big| & & \Big| \end{bmatrix}$$

- b_j is a linear combination of the columns a_k with coefficients c_{kj}

$$c_j = \begin{bmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{mj} \end{bmatrix}$$

清华大学统计学研究中心

57

Examples

58

- Outer product: product of an m-dimensional column vector u with an n-dimensional row vector v

$$\begin{bmatrix} \\ u \\ \\ \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} \Big| & \Big| & & \Big| \\ v_1 u & v_2 u & \cdots & v_n u \\ \Big| & \Big| & & \Big| \end{bmatrix} = \begin{bmatrix} v_1 u_1 & \cdots & v_n u_1 \\ \vdots & & \vdots \\ v_1 u_m & \cdots & v_n u_m \end{bmatrix}$$

rank 1

- $B = AR$

$$\begin{bmatrix} \Big| & & \Big| \\ b_1 & \cdots & b_n \\ \Big| & & \Big| \end{bmatrix} = \begin{bmatrix} \Big| & & \Big| \\ a_1 & \cdots & a_n \\ \Big| & & \Big| \end{bmatrix} \begin{bmatrix} 1 & \cdots & 1 \\ & \ddots & \vdots \\ & & 1 \end{bmatrix} \qquad b_j = Ar_j = \sum_{k=1}^{j} a_k$$

The matrix R is a discrete analogue of an indefinite integral operator

清华大学统计学研究中心

58

A Matrix Inverse Times a Vector

59

- $x = A^{-1}b$
- Think of x as the unique vector that satisfies the equation $Ax = b$
- $A^{-1}b$ is the vector of coefficients of the expansion of b in the basis of columns of A
- Multiplication by A^{-1} is a change of basis operation:

$$\begin{array}{c} b\text{:} \\ \text{coefficients of} \\ \text{the expansion of } b \text{ in} \\ \{e_1,\ldots,e_m\} \end{array} \quad \underset{\text{Multiplication by } A}{\overset{\text{Multiplication by } A^{-1}}{\rightleftarrows}} \quad \begin{array}{c} A^{-1}b\text{:} \\ \text{coefficients of} \\ \text{the expansion of } b \text{ in} \\ \{a_1,\ldots,a_m\} \end{array}$$

清华大学统计学研究中心

59

Multiplication by a Unitary Matrix

60

- A square matrix Q is unitary (in the real case, orthogonal), if $Q^* = Q^{-1}$

  conjugate transpose of Q
- Qx is the linear combination of the columns of Q with coefficients x, and Q^*b is the vector of coefficients of the expansion of b in the basis of columns of Q
- These processes of multiplication by a unitary matrix or its adjoint preserve geometric structure in the Euclidean sense, because inner products are preserved

清华大学统计学研究中心

60

61

Invariance under Unitary Multiplication

- Angles between vectors are preserved, and so are their lengths: $||Qx|| = ||x||$
- In the real case, multiplication by an orthogonal matrix Q corresponds to a rigid rotation (if det$Q$ = 1) or reflection (if det$Q$ = -1) of the vector space

- Theorem. For any $A \in \mathbb{C}^{m\times n}$ and unitary $Q \in \mathbb{C}^{m\times m}$, we have

$$||QA||_2 = ||A||_2 \triangleq \sup_{\substack{x\in\mathbb{C}^n \\ x\neq 0}} \frac{||Ax||_2}{||x||_2}, \quad ||QA||_F = ||A||_F \triangleq \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)^{1/2}$$

清华大学统计学研究中心

61

62

Cholesky Factorization/Decomposition

清华大学统计学研究中心

62

The Object

63

- Hermitian positive definite matrices can be decomposed into triangular factors twice as quickly as general matrices. The standard algorithm for this, Cholesky factorization, is a variant of Gaussian elimination that operates on both the left and the right of the matrix at once, preserving and exploiting symmetry
- In a statistical context, where data is usually real, a 'Hermitian' matrix becomes 'symmetric'

清华大学统计学研究中心

63

Symmetric matrix

64

- For real symmetric matrices we have two crucial properties:
  - All eigenvalues of a real symmetric matrix are real
  - Eigenvectors corresponding to distinct eigenvalues are orthogonal
- So it's possible to choose a complete set of eigenvectors that are perpendicular (or even orthonormal)

清华大学统计学研究中心

64

Hermitian Positive Definite Matrix

65

If A is an $m \times m$ Hermitian positive definite matrix and X is an $m \times n$ matrix of full rank with $m \geq n$, then

- $X^* A X$ is also Hermitian positive definite
- any principal submatrix of A is positive definite
- every diagonal entry of A is a positive real number
- the eigenvalues are also positive real numbers

清华大学统计学研究中心

65

Symmetric Gaussian Elimination

66

- When a single step of Gaussian elimination is applied

$$A = \begin{bmatrix} 1 & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ w & I \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & K - ww^* \end{bmatrix}$$

- In order to maintain symmetry

$$\begin{bmatrix} 1 & w^* \\ 0 & K - ww^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & K - ww^* \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & I \end{bmatrix}$$

- Combining the operations above

$$A = \begin{bmatrix} 1 & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ w & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^* \end{bmatrix} \begin{bmatrix} 1 & w^* \\ 0 & I \end{bmatrix}$$

- The idea of Cholesky factorization is to continue this process, zeroing one column and one row of A symmetrically until it is reduced to the identity

清华大学统计学研究中心

66

Cholesky Factorization

67

- In general, denote $\alpha = \sqrt{a_{11}}$
- $A = \begin{bmatrix} a_{11} & w^* \\ w & K \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ w/\alpha & I \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & K - ww^*/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^*/\alpha \\ 0 & I \end{bmatrix} = R_1^* A_1 R_1$

$\boxed{A_1 = R_1^{-*} A R_1^{-}}$

- The upper-left entry of the submatrix $K - ww^*/a_{11}$ should be positive
- The process is continued down to the bottom-right corner, giving us eventually a factorization

$$A = \underbrace{R_1^* R_2^* \cdots R_m^*}_{R^*} \underbrace{R_m \cdots R_2 R_1}_{R}$$

清华大学统计学研究中心

67

- Theorem. Every Hermitian positive definite matrix A has a unique Cholesky factorization

68

$$A = R^* R, \quad r_{jj} > 0,$$

where R is upper-triangular

- Algorithm:

This algorithm is always stable

Cholesky Factorization

$R = A$
for $k = 1$ to m
    for $j = k + 1$ to m
        $R_{j,j:m} = R_{j,j:m} - R_{k,j:m} R_{kj} / R_{kk}$
    $R_{k,k:m} = R_{k,k:m} / \sqrt{R_{kk}}$

only half of the matrix being operated on needs to be represented explicitly.

- Operation count:

GE: $2/3m^3$

$$\sum_{k=1}^{m} \sum_{j=k+1}^{m} 2(m-j+1)+1 \sim \sum_{k=1}^{m} \sum_{j=k+1}^{m} 2(m-j) \sim 2\sum_{k=1}^{m} \sum_{j=k+1}^{m} j \sim \sum_{k=1}^{m} k^2 \sim \frac{1}{3} m^3 \text{ flops}$$

清华大学统计学研究中心

68

Solution of $Ax = b$

69

- If A is symmetric positive definite, the standard way to solve a system of equations $Ax = b$ is by Cholesky factorization
- Reduces the system to $R^*Rx = b$, then solve two triangular systems in succession:
  - first $R^*y = b$ for the unknown y
  - then $Rx = y$ for the unknown x
- Each triangular solution requires just $\sim m^2$ flops, so the total work is again $\sim 1/3m^3$ flops
- This process is backward stable

清华大学统计学研究中心

69

70

Eigenvalue Decomposition

清华大学统计学研究中心

70

Eigenvalues and Eigenvectors

- Let $A \in \mathbb{C}^{m \times m}$ be a **square** matrix. A nonzero vector $X \in \mathbb{C}^m$ is an eigenvector of A, and $\lambda \in \mathbb{C}$ is its corresponding eigenvalue, if

$$\boldsymbol{Ax = \lambda x}$$

- Idea: the action of a matrix A on a subspace S of $\mathbb{C}^m$ may sometimes mimic scalar multiplication
- When this happens, the special subspace $S = E_\lambda$ is called an eigenspace, and any nonzero $x \in S$ is an eigenvector
- The set of all the eigenvalues of a matrix A is the spectrum of A, a subset of $\mathbb{C}$ denoted by $\Lambda(A)$
- Properties:

$$\det(A) = \prod_{j=1}^{m} \lambda_j, \ \mathrm{tr}(A) = \sum_{j=1}^{m} \lambda_j$$

Geometric and Algebraic Multiplicity

- The dimension of eigenspace E_λ (also the nullspace of A-λI) is known as the geometric multiplicity of λ
- The degree m polynomial $P_A(z) =$ det(zI - A) is called the characteristic polynomial

$$\begin{aligned} p_A(z) &= (z-\lambda_1)(z-\lambda_2)\cdots(z-\lambda_m) \\ &= (z-\lambda_1)^{k_1}(z-\lambda_2)^{k_2}\cdots(z-\lambda_r)^{k_r}, \quad r \le m \end{aligned}$$

- The algebraic multiplicity of an eigenvalue λ_j of A is its multiplicity k_j as a root of P_A

Eigenvalue Problems

73

- Not like the problems involving square or rectangular linear systems of equations
- Make sense only when the range and the domain spaces are the same
- This reflects the fact that in applications, eigenvalues are generally used where a matrix is to be compounded iteratively, either explicitly as a power A^k or implicitly in a functional form such as e^{tA}

清华大学统计学研究中心

73

74

- Broadly speaking, eigenvalue analysis are useful:
  - Algorithmically, simplify solutions of certain problems by reducing a coupled system to a collection of scalar problems
  - Physically, give insight into the behavior of evolving systems governed by linear equations
  - Examples: the study of resonance (e.g., of musical instruments when struck or plucked or bowed) and of stability (e.g., of fluid flows subjected to small perturbations), where eigenvalues tend to be particularly useful for analyzing behavior for large times t

清华大学统计学研究中心

74

Eigenvalue Decomposition

75

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

  here X is nonsingular and Λ is diagonal
- Such a factorization does not always exist!

$$\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}\left[\begin{array}{c|c|c|c} x_1 & x_2 & \cdots & x_m \end{array}\right] = \left[\begin{array}{c|c|c|c} x_1 & x_2 & \cdots & x_m \end{array}\right]\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{bmatrix}$$

- The jth column of X is an eigenvector of A and the jth entry of Λ is the corresponding eigenvalue

清华大学统计学研究中心

75

$Ax= b$

76

- The eigenvalue decomposition expresses a change of basis to "eigenvector coordinates"
- $A = X\Lambda X^{-1}$, $\mathbf{(X^{-1}b) = \Lambda(X^{-1}x)}$
- Thus, to compute Ax, we can expand x in the basis of columns of X, apply Λ, and interpret the result as a vector of coefficients of a linear combination of the columns of X

清华大学统计学研究中心

76

Existence

77

- An eigenvalue whose algebraic multiplicity exceeds its geometric multiplicity is a defective eigenvalue
- A matrix that has one or more defective eigenvalues is a defective matrix
- Theorem. A square matrix has an eigenvalue decomposition $X\Lambda X^{-1}$ if and only if it is nondefective/diagonalizable
- Note: If X is highly ill-conditioned, then a great deal of information may be discarded in passing from A to Λ

清华大学统计学研究中心

77

Unitary Diagonalization

78

- Sometimes not only does an m x m matrix A have m linearly independent eigenvectors, but these can be chosen to be **orthogonal**
- A is unitarily diagonalizable if there exists a unitary matrix Q such that

$$A = Q\Lambda Q^*$$

- A matrix A is normal if $A^*A = AA^*$
- Theorem. A matrix is unitarily diagonalizable if and only if it is normal

清华大学统计学研究中心

78

Schur Factorization

79

- A Schur factorization (unitary triangularization) of a matrix A is a factorization

$$A = QTQ^*$$

where Q is unitary and T is upper-triangular

- Note that since A and T are similar, the eigenvalues of A necessarily appear on the diagonal of T
- Theorem. Every square matrix has a Schur factorization

清华大学统计学研究中心

79

Synopsis: Eigenvalue-revealing Factorizations

80

Factorizations of a matrix that reduce it to a form in which the eigenvalues are explicitly displayed

- A diagonalization $A = X\Lambda X^{-1}$ exists if and only if A is nondefective
- A unitary diagonalization $A = Q\Lambda Q^*$ exists if and only if A is normal
- A unitary triangularization (Schur factorization) $A = QTQ^*$ always exists

清华大学统计学研究中心

80

81

Singular Value Decomposition (SVD)

清华大学统计学研究中心

81

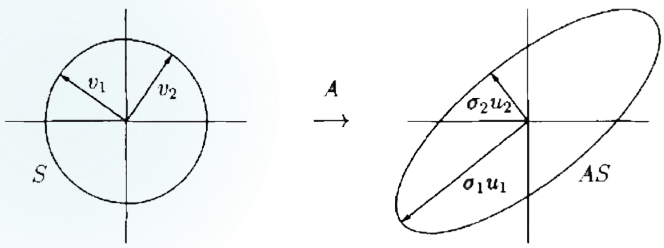
A Geometric Observation

82

- Many problems of linear algebra can be better understood if we first ask the question:

What if we take the SVD?

- Motivation: the image of the unit sphere under any *m* x *n* matrix is a hyperellipse (the m-dimensional generalization of an ellipse)



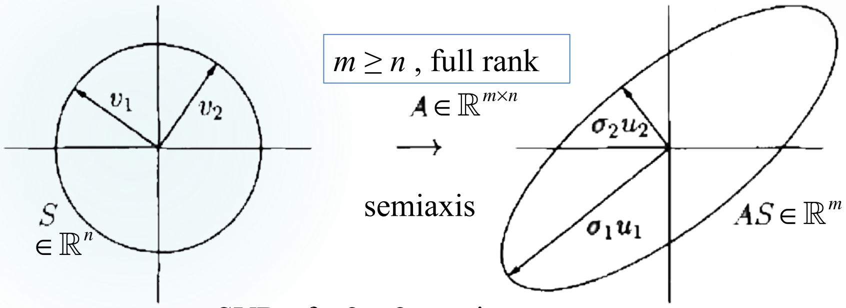
SVD of a 2 x 2 matrix

清华大学统计学研究中心

82

83

- Hyperellipse: the surface obtained by stretching the unit sphere in $\mathbb{R}^m$ by some factors σ_1 , ... , σ_m (possibly zero) in some orthogonal directions u_1, ... , $u_m \in \mathbb{R}^m$
- Take the u_i to be unit vectors, vectors $\{\sigma_i u_i\}$ are the principal semiaxes of the hyperellipse, with lengths σ_i
- Exactly rank(A) of the lengths σ_i will be nonzero



SVD of a 2 x 2 matrix

清华大学统计学研究中心

83

84

Singular Values and Vectors

- The lengths of the n principal semiaxes of AS, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$, are called singular values of A
- The unit vectors $\{u_1, ... , u_n\}$ oriented in the directions of the principal semiaxes of AS, numbered to correspond with the singular values, are called left singular vectors of A
- The right singular vectors of A are the unit vectors $\{v_1, ... , v_n\}$ in S that are the preimages of the principal semiaxes of AS, numbered so that

$$Av_j = \sigma_j u_j, \ 1 \leq j \leq n$$

清华大学统计学研究中心

84

Reduced SVD ($m \geq n$)

85

- $Av_j = \sigma_j u_j,\ 1 \leq j \leq n$

$AV = \hat{U}\hat{\Sigma}$

$$\begin{bmatrix} \\ & A & \\ \\ \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

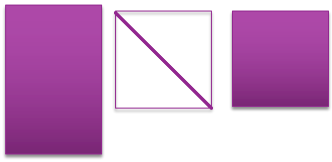
full rank

$n \times n$ matrix with orthonormal columns-unitary

$m \times n$ matrix with orthonormal columns

$n \times n$ diagonal matrix with positive real entries

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



清华大学统计学研究中心

85

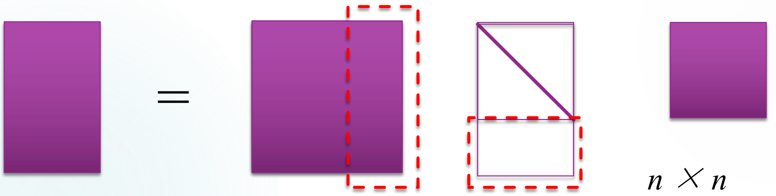
Full SVD

86

- Since $\hat{U} \in \mathbb{C}^m$, extend it to be a unitary matrix

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

"silent" columns/rows



$m \geq n$, full rank

$m \times m$ unitary

$m \times n$ diagonal with positive real entries

$n \times n$ unitary

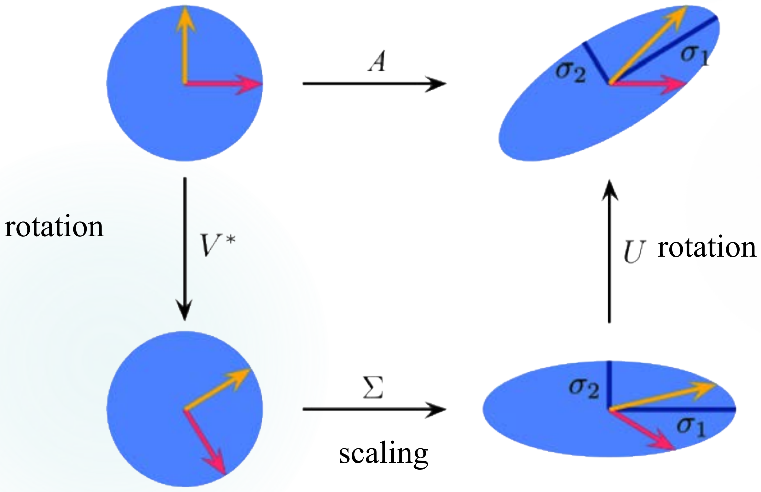
- If rank(A) = $r < n$, extend both U and V to be unitary matrices

清华大学统计学研究中心

86

Visualization

87



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

清华大学统计学研究中心

87

Existence and Uniqueness

88

Theorem

- Every matrix $A \in \mathbb{C}^{m \times n}$ has a singular value decomposition $A = U\Sigma V*$
- Furthermore, the singular values $\{\sigma_j\}$ are uniquely determined; and, if A is square and the σ_j are distinct, the left and right singular vectors $\{u_j\}$ and $\{v_j\}$ are uniquely determined up to complex signs

清华大学统计学研究中心

88

A Change of Bases

- $b = Ax \iff U^*b = U^*U\Sigma V^*x = \Sigma V^*x$

  Denote $b' = U^*b$, $x' = V^*x$, then

  $$\iff \boldsymbol{b' = \Sigma x'}$$

- Thus A reduces to the diagonal matrix Σ when the range(domain) is expressed in the basis of columns of $U(V)$
- The SVD makes it possible for us to say that **every matrix is diagonal**-if only one uses the proper bases for the domain and range spaces

SVD vs. Eigenvalue Decomposition

- For unitarily diagonalizable $A = Q\Lambda Q^*$, this is an ED, and $A = Q|\Lambda|\text{sign}(\Lambda)Q^*$ gives a SVD
- They both have the theme of diagonalizing a matrix, while the ED $A = X\Lambda X^{-1}$ expresses it in terms of one new basis
- $b = Ax = X\Lambda X^{-1}x \iff X^{-1}b = \Lambda X^{-1}x$

  Denote $b' = X^{-1}b$, $x' = X^{-1}x$, then

  $$\iff \boldsymbol{b' = \Lambda x'}$$

  Here both the range and domain are expressed in the basis of eigenvectors of A (columns of X)

91

- Fundamental differences:
  - The SVD uses two different bases (the sets of left and right singular vectors), whereas the ED uses just one (the eigenvectors)
  - The SVD uses orthonormal bases, whereas the ED uses a basis that generally is not orthogonal
  - Not all matrices (even square ones) have an ED, but all matrices (even rectangular ones) have a SVD
- In applications, eigenvalues tend to be relevant to problems involving the behavior of A^k or e^{tA}, whereas singular vectors tend to be relevant to problems involving the behavior of A or A^{-1}

清华大学统计学研究中心

91

Matrix Properties via the SVD

92

- 1. The rank of $A_{m\times n}$ is r, the number of nonzero singular values
- 2. range(A) = $<u_1, \ldots, u_r>$, null(A) = $<v_{r+1}, \ldots, v_n>$
- 3. $||A||_2 = \sup\limits_{\substack{x\in\mathbb{C}^n \\ x\neq 0}} \frac{||Ax||_2}{||x||_2} = \sigma_1, \quad ||A||_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\right)^{1/2} = \sqrt{\sigma_1^2 + \cdots \sigma_r^2}$
- 4. The nonzero singular values of A are the square roots of the nonzero eigenvalues of A^*A or AA^*
- 5. If $A = A^*$, then the singular values of A are the absolute values of the eigenvalues of A
- 6. For $A \in \mathbb{C}^{m\times m}$, $|det(A)| = \prod_{j=1}^{m}\sigma_j$

清华大学统计学研究中心

92

Computational Consequences

93

- The best method for determining the rank of a matrix is to count the number of singular values greater than a judiciously chosen tolerance (Property 1)
- The most accurate method for finding an orthonormal basis of a range or a nullspace is via property 2
- QR factorization provides alternative algorithms that are faster but not always as accurate
- Property 3 represents the standard method for computing $\|A\|_2$
- Besides these examples, the SVD is also an ingredient in robust algorithms for least squares fitting, intersection of subspaces, regularization, and numerous other problems

清华大学统计学研究中心

93

$A = U\Sigma V^*$

94

- U: The left-singular vectors of A are a set of orthonormal eigenvectors of AA^*
- V: The right-singular vectors of A are a set of orthonormal eigenvectors of A^*A
- Σ: The non-zero singular values of A (found on the diagonal entries of Σ) are the square roots of the non-zero eigenvalues of both A^*A and AA^*
- A is the sum of r rank-one matrices:

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^*$$

Low-rank approximations

清华大学统计学研究中心

94

95

Least Squares Problems

清华大学统计学研究中心

95

96

The Problem

- Least squares data-fitting has been an indispensable tool since its invention by Gauss and Legendre around 1800, with ramifications extending throughout the mathematical sciences
- The problem is the solution of an overdetermined system of equations $Ax = b$ - rectangular, with more rows than columns
- The least squares idea is to "solve" such a system by minimizing the 2-norm of the residual $b - Ax$

清华大学统计学研究中心

96

Orthogonal Projection and the Normal Equations

97

- Theorem. Let $A \in \mathbb{C}^{m\times n}(m \geq n)$ and $b \in \mathbb{C}^{m}$ be given. A vector minimizes the residual norm $\|r\|_2 = \|b - Ax\|_2$, thereby solving the least squares problem, if and only if $x \in \mathbb{C}^{n}$, that is, $A^*r = 0$, or equivalently,

$$A^*Ax = A^*b,$$

$\boxed{r \perp \text{range}(A)}$

or again equivalently,

$$Pb = Ax,$$

where $P \in \mathbb{C}^{m\times m}$

is the orthogonal projector onto range(A)

The $n\times n$ system of equations, known as the normal equations, is nonsingular if and only if A has full rank. Consequently the solution x is unique if and only if A has full rank

清华大学统计学研究中心

97

Pseudoinverse

98

- If A has full rank, then the solution x to the least squares problem is given by $x = (A^*A)^{-1}A^*b$
- The pseudoinverse of A is $A^+ = (A^*A)^{-1}A^* \in \mathbb{C}^{n\times m}$
- Then the full-rank linear least squares problem is to compute one or both of the vectors

$$x = A^+b, \quad y = Pb$$

where A^+ is the pseudoinverse of A and P is the orthogonal projector onto range(A)

- There are three leading algorithms for doing this

清华大学统计学研究中心

98

Normal Equations

99

- The classical way is to solve the normal equations
- If A has full rank, this is a square, Hermitian positive definite system of equations of dimension n
- The standard method of solving such a system is by Cholesky factorization, $A^*A = R^*R$, where R is upper-triangular, reducing the normal equations to

$$R^*Rx = A^*b$$

清华大学统计学研究中心

99

Algorithm 1

100

Least Squares via Normal Equations
1. Form the matrix A^*A and the vector A^*b
2. Compute the Cholesky factorization $A^*A = R^*R$
3. Solve the lower-triangular system $R^*w = A^*b$ for w
4. Solve the upper-triangular system $Rx = w$ for x

Operation count:

- To compute A^*A reqires $\sim \sum_{i=1}^{n} \sum_{j=1}^{i} 2m \sim mn^2$ flops
- Cholesky factorization takes $\sim n^3/3$ flops
- Since these two dominate the computation, the total work requires $\sim mn^2+n^3/3$ flops

清华大学统计学研究中心

100

QR Factorization

101

- The "modern classical" method, popular since the 1960s, is based upon reduced QR factorization
- Use Gram-Schmidt orthogonalization or, more usually, Householder triangularization, to get $A = \hat{Q}\hat{R}$
- The orthogonal projector $P = \hat{Q}\hat{Q}^*$, so $y = Pb = \hat{Q}\hat{Q}^*b$
- $Ax = y \leftrightarrow \hat{Q}\hat{R}x = \hat{Q}\hat{Q}^*b \leftrightarrow \hat{R}x = \hat{Q}^*b$

  $A^*Ax = A^*b \leftrightarrow \hat{R}^*\hat{Q}^*\hat{Q}\hat{R}x = \hat{R}^*\hat{Q}^*b \leftrightarrow \hat{R}x = \hat{Q}^*b$

清华大学统计学研究中心

101

Algorithm 2

102

Least Squares via QR Factorization
1. Compute the reduced QR factorization $A = \hat{Q}\hat{R}$
2. Compute the vector $\hat{Q}^*b$
3. Solve the upper-triangular system $\hat{R}x = \hat{Q}^*b$ for x

Operation count:

- The work is dominated by the cost of the QR factorization
- If Householder reflections are used for this step, the total work requires $\sim 2mn^2 - 2/3n^3$ flops

清华大学统计学研究中心

102

SVD and Algorithm 3

103

- Now $A = \hat{U}\hat{\Sigma}V^*$, take $P = \hat{U}\hat{U}^*$, or
  $A^*Ax = A^*b \leftrightarrow V\hat{\Sigma}\hat{U}^*\hat{U}\hat{\Sigma}V^*x = V\hat{\Sigma}\hat{U}^*b \leftrightarrow \hat{\Sigma}V^*x = \hat{U}^*b$

Least Squares via SVD $A = \hat{U}\hat{\Sigma}V^*$

1. Compute the reduced SVD
2. Compute the vector $\hat{U}^*b$
3. Solve the diagonal system $\hat{\Sigma}w = \hat{U}^*b$ for w
4. Set $x = Vw$

Operation count:

- The work is dominated by the cost of the SVD, which requires ~ $2mn^2 + 11\ n^3$ flops

清华大学统计学研究中心

103

Comparison of Algorithms

104

- Each of these methods is advantageous in certain situations
- When speed is the only consideration, Algorithm 1 may be the best
- However, solving the normal equations is not always stable in the presence of rounding errors, thus numerical analysts recommended Algorithm 2 instead
- If A is close to rank-deficient, Algorithm 2 itself has less-than-ideal stability properties, then Algorithm 3 is preferred

清华大学统计学研究中心

104

Homework

- Verify that the flop count is $\sim 3mn^2 - n^3$ for Givens QR on p. 38
- Write R functions for each algorithm, and design examples to compare their results. (CGS, MGS and Householder QR factorization, Gaussian elimination with and without partial pivoting)
- Solve sets of linear equations with your functions
- Let A be a 10 x 10 random matrix with entries from the standard normal distribution, minus twice the identity. Write a program to plot $\|e^{tA}\|_2$ against t for $0 \leq t \leq 20$ on a log scale, comparing the result to the straight line $e^{t\alpha(A)}$, where $\alpha(A) = \max_j \mathrm{Re}\,(\lambda_j)$ is the spectral abscissa of A. Run the program for ten random matrices A and comment on the results. What property of a matrix leads to a $\|e^{tA}\|_2$ curve that remains oscillatory as $t \rightarrow \infty$?

清华大学统计学研究中心