Module – 4.2

MATRIX DECOMPOSITION METHODS

AN OVERVIEW

S. Lakshmivarahan

School of Computer Science

University of Oklahoma

Norman, Ok – 73069, USA

varahan@ou.edu

MATRIX METHODS FOR SOLVING Ax = b

Two classes

- Direct method
- Multiplicative decomposition of A
- Complexity – $O(n^3)$
- Gives exact answer if there is no round – off
- Three decompositions: LU, QR, SVD

- Iterative method
- Additive decomposition of A
- Convergence proof
- Rate of convergence
- Complexity depends on the cost per iteration and the desired accuracy
- Jacobi, Gauss-Seidel, SOR, etc

DIRECT METHOD – LU – DECOMPOSITION OF A

- LU decomposition derived from the classical Gaussian elimination method

- Given A – nonsingular, there exists L, a lower triangular and a U – upper triangular matrices:

$$A = LU$$

LU DECOMPOSITION OF A

- $$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & a_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix}$$

- L has $\frac{n(n-1)}{2}$ unknowns and U has $\frac{n(n+1)}{2}$ unknowns – a total of n^2 unknowns

- Multiplying L and U and equating the elements we can easily solve the system of n^2 equations in n^2 unknowns

EXAMPLE

- A = $\begin{bmatrix} 1 & {}^{3}/_{2} \\ {}^{3}/_{2} & {}^{1}/_{2} \end{bmatrix}$ = $\begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$ = LU

  = $\begin{bmatrix} u_{11} & u_{12} \\ l_{21}u_{11} & l_{21}u_{12} + u_{22} \end{bmatrix}$

- Verify: L = $\begin{bmatrix} 1 & 0 \\ {}^{3}/_{2} & 1 \end{bmatrix}$, U = $\begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & {}^{5}/_{4} \end{bmatrix}$

- By exploiting the patterns in the n^2 nonlinear equations in n^2 unknowns we get the following algorithm for L and U

LU DECOMPOSITION – PSEUDO CODE

- Given $A \in R^{nxn}$, non singular

  For r = 1 to n

  For i = r to n

  $u_{ri} = a_{ri} - \sum_{j=1}^{r-1} l_{rj} u_{ji}$ - Rows of U

  End For

  For i = r + 1 to n

  $l_{ir} = \frac{1}{u_{rr}} [a_{ir} - \sum_{j=1}^{r-1} l_{rj} u_{ji}]$ - Columns of L

  End For

  End For

- Verify that the total number of operation is $O(n^3)$

LU DECOMPOSITION – A FRAME WORK FOR SOLUTION

- Given L, U: A = LU, then

- Ax = (LU)x = L(Ux) = Lg = b and Ux = g

- Summary – a three step procedure
  - Decompose A = LU
  - Solve Lg = b – lower triangular system
  - Solve Ux = g – upper triangular system

SOLUTION LOWER TRIANGULAR SYSTEM: Lg = b

- Let $\begin{bmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$
- Forward elimination method:

  $g_1 = \frac{b_{11}}{l_{11}}$

  For i = 2 to n

  $g_i = \frac{1}{l_{ii}} [b_i - \sum_{j=1}^{i-1} l_{ij} g_j]$

  End For
- Verify that it takes $O(n^2)$ operations to compute g

SOLUTION UPPER TRIANGULAR SYSTEM: Ux = g

- Let $\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}$
- Back substitution method:

  $x_n = \frac{g_n}{u_{nn}}$

  For i = n - 1 to n

  $x_i = \frac{1}{u_{ii}} [g_i - \sum_{j=i+1}^{i-1} u_{ij} x_j]$

  End For
- Verify that it takes $O(n^2)$ operations to compute x

TOTAL CASE OF SOLVING Ax = b

- LU decomposition step – $O(n^3)$

- Lower triangular system – $O(n^2)$

- Upper triangular system – $O(n^2)$

- Total cost is $O(n^3)$

COMPLEXITY OF LARGE PROBLEM

- Let n = 10^6 and $n^3 = 10^{18}$ – operations
- Consider a machine that takes 10^{-12} second per operation. It's a TERA FLOP MACHINE
- TIME needed = $10^{18} x 10^{-12} = 10^6$ seconds
- There are only 60x60x24x365 = 32,536,000 = $31.5x10^6$ seconds in one year
- It takes = $\frac{10^6}{60x60x24}$ = $\frac{10^6}{86,400}$ = 11.575 days to solve Ax = b

WHEN A IS SYMMETRIC

- Let D = diag(u_{11}, u_{22}, … u_{nn}) a diagonal matrix with the diagonal elements of U

- Then U = DM where the diagonal of M are all 1

- Then A = LDM

- If A is symmetric, then M = L^T and A = LDL^T

EXAMPLE

- Recall

$$A = \begin{bmatrix} 1 & {}^{3}/_{2} \\ {}^{3}/_{2} & {}^{1}/_{2} \end{bmatrix} = LU = \begin{bmatrix} 1 & 0 \\ {}^{3}/_{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & {}^{5}/_{4} \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & {}^{5}/_{4} \end{bmatrix} = DM = \begin{bmatrix} 1 & 0 \\ 0 & {}^{5}/_{4} \end{bmatrix} \begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & 1 \end{bmatrix} = L^T \text{ since A is symmetric}$$

$$D^{1/2} = \begin{bmatrix} 1 & 0 \\ 0 & {}^{\sqrt{5}}/_{2} \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 \\ {}^{3}/_{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & {}^{\sqrt{5}}/_{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & {}^{\sqrt{5}}/_{2} \end{bmatrix} \begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ {}^{3}/_{2} & {}^{\sqrt{5}}/_{2} \end{bmatrix} \begin{bmatrix} 1 & {}^{3}/_{2} \\ 0 & {}^{\sqrt{5}}/_{2} \end{bmatrix}$$

$$= GG^T$$

WHEN A – SPD – CHOLESKEY DECOMPOSITION

- When A is PD => diagonal elements of D are positive
- $A = LDL^T = LD^{½}D^{½}L^T$

  $= (LD^{½})(LD^{½})^T$

  $= GG^T$ – Choleskey decomposition
- $G = LD^{½}$ is called the Choleskey factor
- $D^{½} = \mathrm{diag}(u_{11}^{½}, u_{22}^{½}, \ldots\ u_{nn}^{½})$ is the square root of the diagonal matrix D
- G is also known as the square root of A

COMPUTATION OF G GIVEN A

For j = 1 to n

$g_{jj} = [a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2]^{½}$ - diagonal of G

For i = j + 1 to n

$g_{ij} = \frac{1}{g_{jj}}[a_{ij} - \sum_{k=1}^{j-1} g_{ik}\, g_{kj}]$ – column of G

End For

End For

- Verify that it still takes $O(n^3)$ operations but the leading coefficient is one-half of that is required for LU - decomposition

CHOLESKY FRAME WORKL: Ax = b

- A SPD and $A = GG^T$
- $Ax = (GG^T)x = G(G^Tx) = Gy = b$
- Compute G: $A = GG^T$ – $O(n^3)$ operations
- Solve $Gg = b$ – Lower triangular – $O(n^2)$ operations
- Solve $G^Tx = g$ – upper triangular – $O(n^2)$ operations
- Total cost still is $O(n^3)$ with a smaller coefficient in the leading term

SOLUTION OF NORMAL EQUATION: $(H^TH)x = H^TZ$

- Given $H \in R^{mxn}$ of full rank, $Z \in R^m$
- <u>Step 1:</u> Compute H^TH – $O(nm^2)$ operations
- <u>Step 2:</u> Compute H^TZ – $O(nm)$ operations
- <u>Step 3:</u> Compute the cholesky factor G:

  $(H^TH) = GG^T$ – $O(n^3)$ operations
- <u>Step 4:</u> Solve lower triangular system

  $Gg = H^TZ$ – $O(n^2)$ operations
- <u>Step 5:</u> Solve upper triangular system

  $G^Tx = g$ – $O(n^2)$ operations
- Similarly for $(HH^T)y = Z$ and $x = H^Ty$

SQUARE ROOT OF A - SPD

- Three possible definitions of square root of A – SPD

<u>Square root of A</u>

Choleskey factor G

$A = GG^T$

Symmetric square root

$A = s^2$

Eigen decomposition

$A = V\Lambda V^T = \bar{V}\bar{V}^T$

$\bar{V} = V\Lambda^{1/2}$

ORTHOGONAL MATRIX

- <u>FACT:</u> A matrix $A \in R^{nxn}$ is orthogonal if $A^{-1} = A^T$, that is, $A^TA = AA^T = \mathrm{I}$

- Let y = Ax and A be orthogonal. Then

$$\|y\|_2^2 = \|Ax\|_2^2 = (Ax)^T(Ax) = x^TA^TAx = x^Tx = \|x\|_2^2$$

Thus, 2 –norm is <u>invariant</u> under orthogonal transformation

QR – DECOMPOSITION (m > n)

- FACT: Let $H \in R^{m\times n}$. Then exists an orthogonal matrix $Q \in R^{m\times m}$ and an upper triangular matrix $R \in R^{m\times n}$ such that

$$H = QR,\ QQ^T = Q^TQ = I_m$$

$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mn} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1m} \\ q_{21} & q_{22} & \cdots & q_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ q_{m1} & q_{m2} & \cdots & q_{mm} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \\ \hline 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

called the <u>full QR decomposition</u>

- Columns of Q are orthonormal vectors

REDUCED QR – DECOMPOSITION (m > n)

- Let $Q = [Q_1, Q_2]$,

  $Q_1 \in R^{mxn}$ with first n columns of Q

  $Q_2 \in R^{mx(m-n)}$ with the last (m – n) columns of Q

- $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$

  $R_1 \in R^{nxn}$ with first n columns of R

  $R_2 \in R^{m-nxn}$ is a zero matrix

- Then $H = QR = [Q_1, Q_2]\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = Q_1R_1$ called reduced QR decomposition
- $Q_1^T Q_1 = I_n$

LINEAR LEAST SQUARE PROBLEM: Z = Hx

- r(x) = Z – Hx – residual

- f(x) = $\|\mathrm{r}(x)\|_2^2 = \|Q^T \mathrm{r}(\mathrm{x})\|_2^2 = \|Q^T(\mathrm{Z} - Hx)\|_2^2$ – (Q – orthogonal)

  = $\|Q^T\mathrm{Z} - Q^T Hx\|_2^2$

- Q^TZ = $\begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \mathrm{Z} = \begin{bmatrix} Q_1^T Z \\ Q_2^T Z \end{bmatrix}$
- Q^THx = Q^TQRx = Rx = $\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$x = $\begin{bmatrix} R_1 \mathrm{x} \\ 0 \end{bmatrix}$
- f(x) = $\left\|Q_1^T \mathrm{Z} - R_1 x\right\|_2^2 + \|Q_2 Z\|_2^2$

LEAST SQUARE SOLUTION – QR METHOD

- f(x) = $\left\| Q_1^T \mathrm{Z} - R_1 \mathrm{x} \right\|_2^2 + \|Q_2 Z\|_2^2$

- Only the first term depends on x

- f(x) is a minimum when $R_1x = \mathrm{Q}_1^{\mathrm{T}}\mathrm{Z}$

- $x_{LS} = \mathrm{R}_1^{-1}(\mathrm{Q}_1^{\mathrm{T}}\mathrm{Z})$ is obtained by solving an upper triangular system

QR DECOMPOSITION: m < n

- $Z = Hx$, $H \in R^{mxn}$, $m < n$

- Then $H^T = QR$ as above, since $n > m$

  with $Q = [Q_1, Q_2]$, $R = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}$, $Q_1 \in R^{nxm}$, $Q_2 \in R^{nx(n-m)}$

  $R_1 \in R^{mxm}$ and $R_2 \in R^{n-mxm}$ is a zero matrix

- $Q_1^T Q_1 = I_m$ and $H = R^T Q^T$

LEAST SQUARE SOLUTION – QR METHOD (m < n)

- $f(x) = \|r(x)\|_2^2 = (Z - R^TQ^Tx)^T(Z - R^TQ^Tx)$

  $= Z^TZ - 2Z^TR^TQ^Tx + x^T(QRR^TQ^T)x$

- $\nabla_x f(x) = -2QRZ + 2(QRR^TQ^T)x = 0$

- $\nabla_x^2 f(x) = 2QRR^TQ^T$

- x_{LS} is the solution of: $RR^TQ^TQ = RZ$

FORM OF THE LEAST SQUARE SOLUTION

- $y = Q^T x = \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} x = \begin{bmatrix} Q_1^T x \\ Q_2^T x \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ $y_1 \in R^m$, $y_2 \in R^{n-m}$

- $RR^{T+} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} [R_1^T \; : 0 \,] = \begin{bmatrix} R_1 R_1^T & 0 \\ 0 & 0 \end{bmatrix}$

- $\begin{bmatrix} R_1 R_1^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} R_1 Z \\ 0 \end{bmatrix}$ => $R_1 R_1^T y_1 = R_1 Z$, y_2 is arbitrary

- Y_1 is obtained by solving a lower triangular system $R_1^T y_1 = Z$

THE LEAST SQUARE SOLUTION: m < n

- X = Qy = Q$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ = [Q_1 Q_2] $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ = $Q_1y_1 + Q_2y_2$

- Since y_2 is arbitrary, there are infinitely many solutions

- Clearly, $x_{LS} = Q_1y_1 = Q_1(R_1^{-T}\mathrm{Z})$

- $\|x\|_2^2 = \|Q_1y_1\|_2^2 + \|Q_2y_2\|_2^2$ $\qquad$ ($Q_1^T Q_1 = \mathrm{I}_m$, $Q_1^T Q_1 = \mathrm{I}_{n\text{-}m}$)
  $= \|y_1\|_2^2 + \|y_2\|_2^2$
  $\geq \|y_1\|_2^2 = \|x_{LS}\|_2^2$

SUMMARY: QR ALGORITHM

- Over determined case $H \in R^{mxn}$, m > n

- Step 1: Compute $Q_1 \in R^{mxn}$ and $R_1 \in R^{nxn}$ such that $H = Q_1R_1$ using Gramm-Schmidt orthogonalization method – See below

- Step 2: Compute $Q_1^T Z$

- Step 3: Solve upper triangular system $R_1x = Q_1^T Z$ and $x_{LS} = R_1^{-T}(Q_1^T Z)$

SUMMARY: QR ALGORITHM

- Under determined case $H \in R^{mxn}$, $m < n$
- Step 1: Compute $H^T = Q_1R_1$, $Q_1 \in R^{nxm}$ and $R_1 \in R^{mxm}$
- Step 2: Solve the lower triangular system $R_1^T y_1 = Z$
- Step 3: $x_{LS} = Q_1y_1 = Q_1(R_1^{-T}Z)$

GRAMM- SCHMIDT ORTHOGONALIZATION

- Let H = [h_1, h_2, … h_n], $h_i \in R^m$, $1 \le i \le n$, $m > n$

- Let the columns of are linearly independent

- Find Q = [q_1, q_2, … q_n], $q_i \in R^m$, $1 \le i \le n$ and $\{q_i\}_{i=1}^{n}$ is an orthogonal system:

$$q_i^T q_j = 0 \quad i \neq j$$

$$= 1 \text{ if } i = j$$

- Problem: Given $\{h_i\}_{i=1}^{n}$, find $\{q_i\}_{i=1}^{n}$ with the above properties

ALGORITHM – AN IDEA

- Set $q_1 = \frac{h_1}{r_{11}}$ with $r_{11} = \|h_1\|_2$ and $\|q_1\| = 1$
- Set $q_2 = \frac{1}{r_{22}}[h_2 - r_{12}q_1]$ – 2 unknowns: r_{12}, r_{22}

  Thus, $0 = q_1^T q_2 = \frac{1}{r_{22}}[q_1^T h_2 - r_{12}]$

  Therefore, $r_{12} = q_1^T h_2$ and $r_{22} = \|h_2 - r_{12}q_1\|$
- <u>In general</u>: j – unknowns ($1 \leq j \leq n$)

  $$q_j = \frac{1}{r_{jj}}[h_j - \sum_{i=1}^{j-1} r_{ij}q_i]$$

  $\Rightarrow$ $r_{ij} = q_i^T h_j$ $1 \leq i \leq j-1$

  $$r_{ji} = \left\| h_j - \sum_{i=1}^{j-1} r_{ij}q_i \right\|$$

QR – ALGORITHM – PSEUDO CODE

- Given $\{h_1, h_2, h_3, \dots h_n\}$, $h_i \in R^m$, $m > n$ linearly independent
- Find $\{q_1, q_2, q_3, \dots q_n\}$, $h_i \in R^m$, orthonormal

<u>Step 1:</u> Repeat the following steps 2 to 5 for $j = 1$ to n

<u>Step 2:</u> $v_j = h_j$

<u>Step 3:</u> For $i = 1$ to $j - 1$

Compute: $r_{ij} = q_i^T h_j$

Update: $v_j = v_j - r_{ij} q_i$

<u>Step 4:</u> Compute norm of v_j: $r_{ij} = ||v_j||$

<u>Step 5:</u> $q_j = \frac{v_j}{r_{jj}}$

SINGULAR VALUE DECOMPOSITION - SVD

- Let $H \in R^{mxn}$ be of full rank

Grammians ($m > n$)

- $H^TH \in R^{nxn}$
- Rank (H^TH) = n
- Symmetric
- Positive definite

- $HH^T \in R^{mxm}$
- Rank (HH^T) = m
- Symmetric
- Positive semi-definite

- Let (λ_i, v_i) be the n-eigenvalue eigenvector pair for H^TH with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > 0$
- $(H^TH)v_i = \lambda_i v_i \qquad 1 \leq i \leq n$
- $V = [v_1, v_2, \ldots v_n]$, $\Lambda = \text{Diag}[\lambda_1, \lambda_2, \ldots \lambda_n]$
- V is orthogonal matrix, $V^TV = VV^T = I$
- $(H^TH)V = V\Lambda$ or $(H^TH) = V\Lambda V^T$

EIGENVALUES AND VECTORS OF HH^T

- Define $u_i = \frac{1}{\sqrt{\lambda_i}} H v_i$, $u_i \in R^m$, $1 \leq j \leq n$

- $(HH^T)u_i = (HH^T)\frac{1}{\sqrt{\lambda_i}} H v_i$

$$= \frac{1}{\sqrt{\lambda_i}} H(H^T H) v_i$$

$$= \frac{1}{\sqrt{\lambda_i}} H \lambda_i v_i = \sqrt{\lambda_i} H v_i = \lambda_i u_i$$

- Thus, (λ_i, u_i), $1 \leq j \leq n$ are the eigenvectors of HH^T

- The rest of (m-n) eigenvalues of (HH^T) are zeros

EIGENDECOMPOSTION OF HH^T

- Set $U = [u_1, u_2, \ldots u_n] \in R^{mxn}$
- $u_i = \frac{1}{\sqrt{\lambda_i}} H v_i$ $\Rightarrow U\Lambda^{½} = HV$
- $U^T U = (HV\Lambda^{-½})^T (HV\Lambda^{-½})$

  $= \Lambda^{-½} V^T (H^T H) V \Lambda^{-½}$

  $= \Lambda^{-½} V^T V \Lambda \Lambda^{-½}$

  $= I$ (because $V^T V = I$)
- Columns of U are orthonormal

SVD OF H

- $u_i = \frac{1}{\sqrt{\lambda_i}} H v_i$ => $H v_i = u_i \sqrt{\lambda_i}$
- $HV = U\Lambda^{½}$ or $H = U\Lambda^{½}V^T$ is called the SVD of H
- $H = [u_1, u_2, \ldots u_n] \begin{bmatrix} \lambda_1^{1/2} & 0 & \cdots & 0 \\ 0 & \lambda_2^{1/2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n^{1/2} \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}$
- $H = \sum_{i=1}^{n} \sqrt{\lambda_i} u_i v_i^T$
- λ_i are eigenvalues of H^T and $\lambda_i^{1/2}$ are the singular values of H by definition

SVD BASED SOLUTION OF LEAST SQUARES

- $Z = Hx$, $H \in R^{mxn}$ – full rank
- $H = U\Lambda^{½}V^T$, $VV^T = V^TV = I_n$, $U^TU = I_n$
- $f(x) = (Z - Hx)^T(Z - Hx)$

  $= (Z - U\Lambda^{½}V^Tx)^T(Z - U\Lambda^{½}V^Tx)$

  $= Z^TZ - 2Z^TU\Lambda^{½}V^Tx + x^T(V\Lambda V^T)x$
- $0 = \nabla_x f(x) = -2V\Lambda^{½}U^TZ - 2(V\Lambda V^T)x$
- x_{LS} is the solution of: $(V\Lambda V^T)x = V\Lambda^{½}U^TZ$
- $x_{LS} = V\Lambda^{-½}U^TZ$

ALGORITHM - SVD

- Given $H \in R^{mxn}$

STEP 1: Compute $H = U\Lambda^{½}V^T$

STEP 2: Compute $U^T Z$ – (rotation)

STEP 3: Compute $y = \Lambda^{-½}U^T Z$ – (Scaling)

STEP 4: Compute $x^* = Vy$ – (rotation)

EXERCISES

13.1) Consider the matrix $H \in R^{4x16}$ built using the 2-D bilinear interpolation in Module 3.6

(a) Pick for pairs (a_i, b_i), $1 \leq i \leq 4$ of uniformly distributed random numbers in range [0, 1]

(b) Compute the elements of the rows of H and verify that they add up to 1

(c) Compute HH^T

(d) Generate observation $Z_i = 75 + V_i$, $V_i \sim N(0, \sigma^2)$ for $1 \leq i \leq 4$

EXERCISES

13.2) Develop your own MATLAB program to do the following

(a) LU - decomposition
(b) Solving lower and upper triangular system
(c) Cholesy decomposition
(d) Gramm–Schmidt orthogonalization
(e) SVD

EXERCISES

13.3)

(a) Apply Cholesky decomposition to solve $(HH^T)y = Z$ and compute $x_{LS} = H^T y$

(b) Compute $\hat{Z} = Z - Hx_{LS}$ and $r(x_{LS}) = Z - \hat{Z}$. Compute $\|r(x_{LS})\|_2$

13.4) Apply QR Decompostion to H using Gramm-Schmidt and slove the resulting linear least square problem

13.5) Apply SVD to H and solve the resulting least square problem

13.6) Compare the norm of the residual $r(x) = Z - \hat{Z}$ computed using the three methods

REFERENCES

- This module follows closely the developments in chapter 9 of LLD (2006)
- Basic iterative methods for solving Ax = b are covered in the following books:
  - G. H. Golub and C. F. Van Loan (1989) Matrix Computation, *Johns Hopkins University Press (Second Edition)*
  - Hageman, L. A. and D. Young (1981) Applied Iterative Methods, *Academic Press*