Lecture 7: Givens Orthogonalization Methods, Weighted Least Squares and Computation of Pseudo Inverse

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6435

Adv Numerical Methods in Sci Comp

Fall 2008
October 8, 2008

Copyright © 2008 by K. Pattipati

Outline of Lecture 7

- Givens Transformations
- Weighted Least Squares Problem and its Solutions via Householder Transformation
- Computation of Pseudo (Generalized) Inverse

Copyright © 2008 by K. Pattipati

How do Given's Rotations Work? - 1

- Selective zeroing of elements and selective revision of R

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

$$\xrightarrow{(1,4)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & x & x \\ x & x & x \end{bmatrix} \xrightarrow{(2,4)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & x \\ x & x & x \end{bmatrix} \xrightarrow{(3,4)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ x & x & x \end{bmatrix} \xrightarrow{(1,5)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & x & x \end{bmatrix}$$

 Copyright © 2008 by K. Pattipati

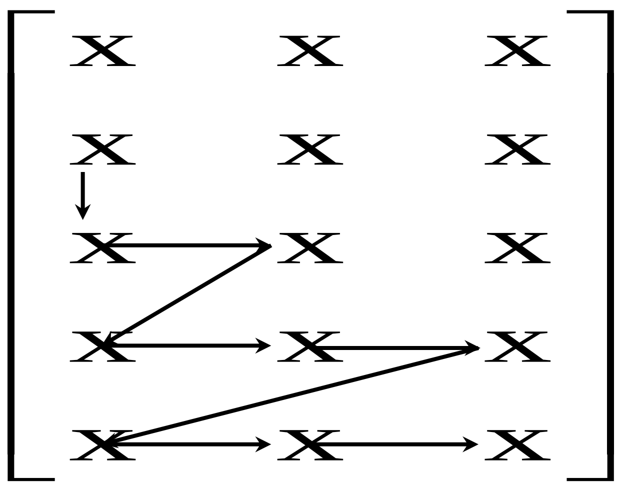
How do Given's Rotations Work? - 2

$$\xrightarrow{(2,5)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & x \end{bmatrix} \xrightarrow{(3,5)} \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$J_{(3,5)}J_{(2,5)}J_{(1,5)}J_{(3,4)}J_{(2,4)}J_{(1,4)}J_{(2,3)}J_{(1,3)}J_{(1,2)}A=R$$

$$\Rightarrow Q^T = J_{(3,5)}J_{(2,5)}J_{(1,5)}J_{(3,4)}J_{(2,4)}J_{(1,4)}$$

- Zig-zag pattern of zeroed-out elements



Zero out elements
$j \le \min(i\text{-}1,n)$ in row i

 Copyright © 2008 by K. Pattipati

Given's Transformation (Rotations)

- What are these Givens rotations ?

$$J(i,k,\theta) = \begin{array}{c} \\ \\ i \\ k \\ \\ \end{array} \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ .. & .. & c & s & 0 \\ .. & .. & -s & c & 0 \\ 0 & .. & .. & .. & 1 \end{bmatrix}$$

$$\qquad\qquad\qquad i \quad k$$

c=cosθ

s=sinθ

$$= I + (\underline{v}_1 - \underline{e}_i)\underline{e}_i^T + (\underline{v}_2 - \underline{e}_k)\underline{e}_k^T$$

where, $\underline{v}_1 = \begin{bmatrix} 0 \\ 0 \\ : \\ c \\ : \\ -s \\ : \\ 0 \end{bmatrix}$ and $\underline{v}_2 = \begin{bmatrix} 0 \\ 0 \\ : \\ s \\ : \\ c \\ : \\ 0 \end{bmatrix}$ Note: $\underline{v}_1^T \underline{v}_2 = 0$

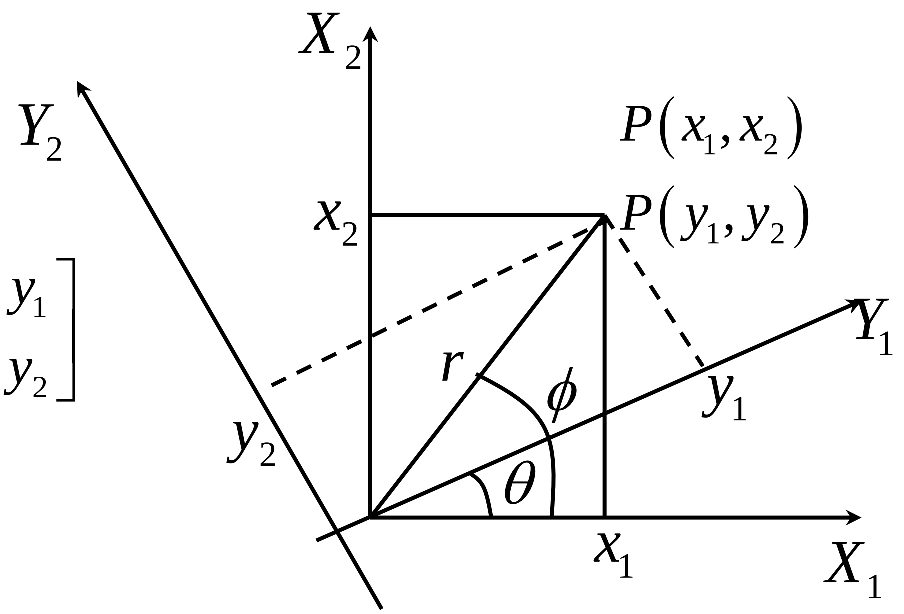
$\Rightarrow J(i,k,\theta)$ is a rank two correction to an identity matrix

Copyright © 2008 by K. Pattipati

Why is Givens Transformation a Rotation?

- To motivate Givens rotations, consider the two-dimensional case:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} cx_1 + sx_2 \\ -sx_1 + cx_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$



$$x_1 = r\cos\phi, x_2 = r\sin\phi$$

$$y_1 = r\cos(\phi-\theta) = r\cos\phi\cos\theta + r\sin\phi\sin\theta$$

$$y_2 = r\sin(\phi-\theta) = r\sin\phi\cos\theta - r\cos\phi\sin\theta$$

Copyright © 2008 by K. Pattipati

Why is Givens Transformation a Rotation?

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}}_{J(1,2,\theta)\text{ matrix}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow \text{Rotation of X-Y axis through an angle } \theta$$

Also, $$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}_{J(1,2,-\theta)\text{ matrix}} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

- So, we have the important result that

$$J^{-1}(1,2,\theta) = J^T(1,2,\theta) = J(1,2,-\theta)$$

- In general, $J(i,k,\theta)$ rotates i-k coordinates by an angle θ in a counter-clockwise direction.

$$J(i,k,\theta)\underline{x} = \underline{y}$$

$$\Rightarrow y_i = cx_i + sx_k, \quad y_k = -sx_i + cx_k, \quad y_j = x_j \forall j \neq i,k$$

Copyright © 2008 by K. Pattipati

Householder versus Givens

- Also, note in 2 by 2 case, if $\underline{v} = [v_1, v_2]^T$ where, $v_1 = -\sin\theta,\ v_2 = \cos\theta$

$$W = \left[I - 2\underline{v}\underline{v}^T\right] = \begin{bmatrix} 1-2v_1^2 & -2v_1v_2 \\ -2v_1v_2 & 1-2v_2^2 \end{bmatrix} = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}$$

- For a 2 by 2 case

$$Q = \begin{bmatrix} c & s \\ s & -c \end{bmatrix}$$ is a Householder (or reflection)

$$Q = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$ or $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$ is a rotation

- Coming back to the general case, we can force $y_k \uparrow$ to 0 by letting

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}} \qquad s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}}$$

$\Rightarrow$ Any specified element can be <u>zeroed out</u> by appropriate choice of c and s

$\Rightarrow$ Since the effect is local, the procedure is well-suited for parallel processing

Copyright © 2008 by K. Pattipati

Implementation Issues

$$|x_k| \geq |x_i|, \text{ write } t = x_i / x_k; s = \left(1+t^2\right)^{-1/2}, c = st$$

$$|x_i| \geq |x_k|, \text{ write } t = x_k / x_i; c = \left(1+t^2\right)^{-1/2}, s = ct$$

- Implementation

if $x_k = 0$

$c = 1$

$s = 0$

else if $|x_k| \geq |x_i|$

$$t = x_i / x_k; s = \left(1+t^2\right)^{-1/2}, c = st$$

else

$$t = x_k / x_i; c = \left(1+t^2\right)^{-1/2}, s = ct$$

end if

Copyright © 2008 by K. Pattipati

Local Effect of Givens

- What is $J(i,k,\theta)A$ and $AJ(i,k,\theta)$?

$$\left.\begin{array}{l} J(i,k,\theta)A \text{ affects only rows } i \text{ and } k \text{ of } A \\ AJ(i,k,\theta) \text{ affects only columns } i \text{ and } k \text{ of } A \end{array}\right\} \text{Local effect}$$

$\underline{J(i,k,\theta)A}$	$\underline{AJ(i,k,\theta)}$
For $j=1,2,\ldots,n$ DO	For $i=1,2,\ldots,m$ DO
$v=a_{ij}$	$v=a_{li}$
$w=a_{kj}$	$w=a_{lk}$
$a_{ij}=cv+sw$	$a_{li}=cv+sw$
$a_{kj}=-sv+cw$	$a_{lk}=-sv+cw$
End DO	End DO
$O(2n)$ operations	$O(2m)$ operations

Copyright © 2008 by K. Pattipati

Givens Orthogonalization Procedure

Algorithm: Givens

For $k = 2,\ldots,m$ DO

For $i = 1,2,\ldots,\min(k-1,n)$ DO

Find c and s ∍

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}\begin{bmatrix} a_{ii} \\ a_{ki} \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}$$

$$A \leftarrow J(i,k,\theta)A$$

End DO

End DO

- Number of operations: $2n^2\left(m - \frac{n}{3}\right)$
- If you want to solve LS problem, insert $\underline{b} \leftarrow J(i,k,\theta)\underline{b}$ and solve $R\underline{x} = \underline{b}$

Copyright © 2008 by K. Pattipati

Iterative Improvement of LS Solution

- Assume Full Column Rank. Suppose we have an initial solution $\underline{x}_{LS}^{0}$

  Can we improve it ? YES !!

Iteration k=0

$+\rightarrow$ Compute residual $\underline{r}^{(k)} = \underline{b} - A\underline{x}_{LS}^{(k)}$ in double precision

| Solve LS problem $\min_{\underline{z}^{(k)}} \left\| A\underline{z}^{(k)} - \underline{r}^{(k)} \right\|_2^2$

| $\Rightarrow$ solve $R_1 \underline{z}^{(k)} = r_C$

| where r_C is given by

$$Q^T \underline{r}^{(k)} = \begin{bmatrix} r_C \\ r_d \end{bmatrix} \begin{matrix} n \\ m-n \end{matrix}$$

| $\underline{x}_{LS}^{(k+1)} = \underline{x}_{LS}^{(k)} + \underline{z}^{(k)}$

| If $\underline{x}_{LS}$ has converged, stop.

| else

| $k = k + 1$

$+\rightarrow$ endif

Computational load: $O(mn + n^2/2)$

Copyright © 2008 by K. Pattipati

Weighted Least Squares (WLS) Problem

- Suppose measurements are noisy and noise process is correlated.

$$\underline{b} = A\underline{x} + \underline{v}$$

$\underline{v}$ – zero mean stochastic process with $E\{\underline{v}\,\underline{v}^T\} = V = SS^T$;

S – Cholesky triangle

Solve weighted least squares (WLS) problem:

$$\min_{\underline{x}} (A\underline{x} - \underline{b})^T V^{-1} (A\underline{x} - \underline{b}) = \left\| S^{-1}(A\underline{x} - \underline{b}) \right\|_2^2 \qquad (1)$$

- One way: Form $S^{-1}A = \bar{A}$ and $S^{-1}\underline{b} = \bar{\underline{b}}$ and solve ordinary least squares problem. $\min_{\underline{x}} \left\| \bar{A}\underline{x} - \bar{\underline{b}} \right\|_2^2$

  - If S is ill-conditioned, $\underline{x}_{\text{LS}}^0$ is bad !!

Copyright © 2008 by K. Pattipati

Optimality Conditions of WLS Problem

- Better way: What are the necessary conditions of optimality ?

$$\left(A^T V^{-1} A\right)\underline{x} = A^T V^{-1}\underline{b}$$

$$V = SS^T \Rightarrow V^{-1} = (S^{-1})^T S^{-1}$$

$$\underline{x}_{\text{LS}} = \left(A^T V^{-1} A\right)^{-1} A^T V^{-1}\underline{b} \quad \text{when } A \text{ is full rank}$$

$$= \left[\left(S^{-1}A\right)^T S^{-1}A\right]^{-1}\left(S^{-1}A\right)^T\left(S^{-1}\underline{b}\right)$$

To derive an efficient method, let us look at an alternate problem:

$$\min \frac{1}{2}\underline{v}^T\underline{v}$$

$$\text{s.t. } \underline{b} = A\underline{x} + S\underline{v} \qquad (2)$$

 Copyright © 2008 by K. Pattipati

Necessary Conditions of Optimality

$$L = \frac{1}{2}\underline{v}^T\underline{v} + \lambda^T\left[A\underline{x} + S\underline{v} - \underline{b}\right]$$

$$\Rightarrow \partial \mathrm{L}/\partial \underline{v} = \underline{0} \Rightarrow v + S^T\underline{\lambda} = \underline{0}$$

$$\partial \mathrm{L}/\partial \underline{x} = \underline{0} \Rightarrow A^T\underline{\lambda} = \underline{0}$$

$$V = SS^T \Rightarrow V^{-1} = (S^{-1})^T S^{-1}$$

$$\partial \mathrm{L}/\partial \underline{\lambda} = \underline{0} \Rightarrow \underline{b} = A\underline{x}_{\mathrm{LS}} + S\underline{v}$$

$$\text{or, } \underline{b} = A\underline{x}_{\mathrm{LS}} - SS^T\underline{\lambda} \Rightarrow \underline{\lambda} = V^{-1}\left(A\underline{x}_{\mathrm{LS}} - \underline{b}\right)$$

and $\underline{v} = S^{-1}\left(\underline{b} - A\underline{x}_{\mathrm{LS}}\right)$ weighted residual or "whitened residual"

using $A^T\underline{\lambda} = \underline{0}$, we have

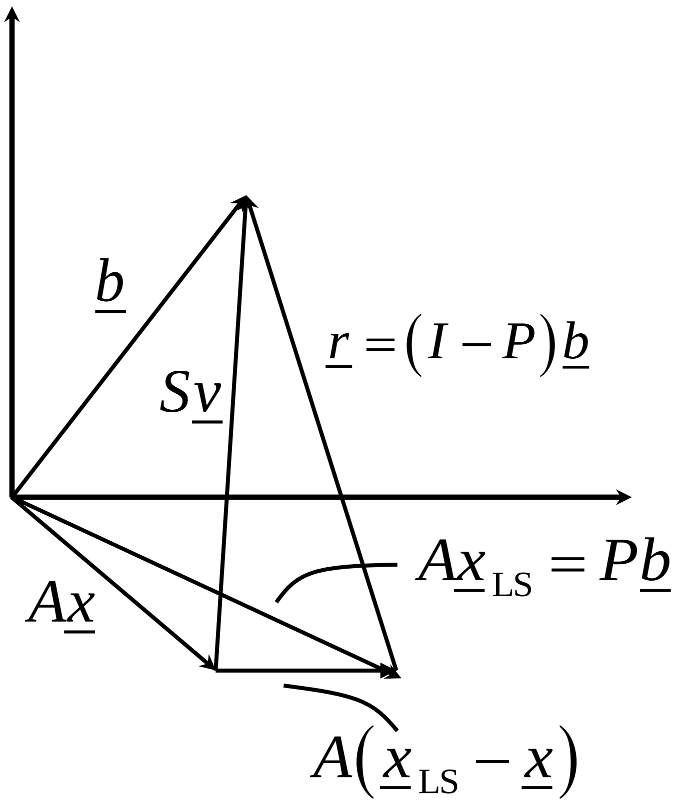
$$\underline{x}_{\mathrm{LS}} = \left(A^T V^{-1} A\right)^{-1} A^T V^{-1}\underline{b}$$

$\Rightarrow$ Problems 1 and 2 are equivalent

$\Rightarrow$ Valid for rank deficient case as well

 Copyright © 2008 by K. Pattipati

Geometric Interpretation



$$\underline{b} = A\underline{x} + S\underline{v}$$
$$= A\underline{x}_{LS} + \underline{r}$$
$$A\underline{x}_{LS} = A\left(A^T V^{-1} A\right)^{-1} A^T V^{-1} \underline{b} = P\underline{b}$$
$$\underline{r} = \underline{b} - A\underline{x}_{LS} = (I - P)\underline{b}$$
$$P = \text{ Projection matrix } (P^2 = P)$$

$$\Rightarrow \boxed{A\left(\underline{x}_{LS} - \underline{x}\right) + \underline{r} = S\underline{v}}$$

Copyright © 2008 by K. Pattipati

Solution via Householder

- Do Householder on A

$$\Rightarrow Q^T A = \begin{matrix} n \\ m-n \end{matrix}\begin{bmatrix} R_1 \\ 0 \end{bmatrix} \quad ; Q = m\begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \quad ; R_1 \text{ upper } \Delta$$

$$\quad n \qquad\qquad\qquad n \quad m-n$$

- Form $Q_2^T S$ and find orthogonal transformation P such that

$$\Rightarrow Q_2^T SP = m-n\begin{bmatrix} 0 & U \end{bmatrix} ; P = m\begin{bmatrix} P_1 & P_2 \end{bmatrix} ; U \text{ upper } \Delta = Q_2^T SP_2$$

$$n \; m-n \qquad\qquad n \quad m-n$$

❑ What does $A\underline{x} + S\underline{v} = \underline{b}$ mean ?

$$\begin{bmatrix} Q_1^T \underline{b} \\ Q_2^T \underline{b} \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \underline{x} + \begin{bmatrix} Q_1^T SP_1 & Q_1^T SP_2 \\ 0 & U \end{bmatrix} \begin{bmatrix} P_1^T \underline{v} \\ P_2^T \underline{v} \end{bmatrix}$$

so solve:

$$U\underline{z} = Q_2^T \underline{b} \qquad \text{and} \qquad \underline{v} = P_2 \underline{z}$$

$$R_1 \underline{x} = Q_1^T \underline{b} - \left(Q_1^T SP_1 P_1^T \underline{v} + Q_1^T SP_2 P_2^T \underline{v} \right)$$

$$= Q_1^T \underline{b} - \left(Q_1^T S\underline{v} \right) = Q_1^T \left[\underline{b} - SP_2 \underline{z} \right]$$

Copyright © 2008 by K. Pattipati

$A^{\dagger}$: Computation of Pseudo Inverse

Solution of $A\underline{x} = \underline{b}$ for any m x n matrix of any rank r

- When m > n and rank(A) = n

$$\underline{x}_{\text{LS}} = \left(A^T A\right)^{-1} A^T \underline{b} = A^{\dagger}\underline{b}$$

- We can always consider $m > n$. Otherwise, use A^T and note $(A^T)^{\dagger} = (A^{\dagger})^T$
- More generally, $\underline{x}_{\text{LS}} = A^{\dagger}\underline{b}$; $A^{\dagger} \sim$ **Moore-Penrose Inverse**, an n x m matrix. Variously referred to as pseudo-inverse or generalized inverse

❑ $A^{\dagger}$ satisfies the following four conditions, termed the **Moore-Penrose conditions**:

i. $A A^{\dagger} A = A$

ii. $A^{\dagger} A A^{\dagger} = A^{\dagger}$

iii. $(A^{\dagger}A) = (A^{\dagger}A)^T \Rightarrow$ projection on to $R(A^T)$

iv. $(AA^{\dagger}) = (AA^{\dagger})^T \Rightarrow$ projection on to $R(A)$

Copyright © 2008 by K. Pattipati

Moore-Penrose Conditions

- Note that ordinary inverse satisfies Moore-Penrose conditions
- $(A^TA)^{-1}A^T$ satisfies conditions
- A full column rank $\Rightarrow A = QR$ and $A^\dagger = R^{-1}Q^T$
- We can show that all four conditions are satisfied by $A^\dagger = R^{-1}Q^T$
  - $QRR^{-1}Q^TQR = QR = A$
  - $R^{-1}Q^TQR\,R^{-1}Q^T = R^{-1}Q^T = A^\dagger$
  - $QR\,R^{-1}Q^T = I_m = I_m^T$
  - $R^{-1}Q^TQR = I_n = I_n^T$

Gram-Schmidt procedure to compute $A^\dagger$

- Suppose A has rank r. Further suppose that A is partitioned as follows:

  $A = [R\ \ T] = [R\ \ RS]$ where $S = t$ x $(n-r)$ matrix

  R has r independent columns, T has $(n\text{-}r)$ dependent columns

  => $\underline{t}_i = R\underline{s}_i,\ i = 1,2,\ldots,n-r$

  => Since R has linearly independent columns

  $R^\dagger = (R^TR)^{-1}R^T$ an r x n generalized inverse matrix

  Also, $R^\dagger R = I_r$ an r x r identity matrix

Copyright © 2008 by K. Pattipati

Gram-Schmidt Procedure for Computing A†

- **Fact:** Generalized inverse of a general *A* is similar to the generalized inverse of [*R* | *RS*], since [*R* | *RS*] can be obtained by the permutation of the columns of *A*. That is,

$$A P_1 P_2 \ldots P_r = A P = [\, R \mid RS \,]$$
$$(AP)^{\dagger} = P^{-1} A^{\dagger} = P^{T} A^{\dagger} = P_r P_{r-1} \ldots P_1 [\, R \mid RS \,]^{\dagger}$$

$\Rightarrow A^{\dagger} = P_1 P_2 \ldots P_r [\, R \mid RS \,]^{\dagger}$ i.e., do a row permutation on the pseudo-inverse of $[R \mid RS]$ in reverse order to obtain the pseudo-inverse of *A*.

- **How to compute [R | RS] †**

claim: $$(AP)^{\dagger} = [R \mid \mathrm{RS}]^{\dagger} = \begin{matrix} r \\ n-r \end{matrix} \begin{bmatrix} (I_r + SS^T)^{-1} R^{\dagger} \\ S^T (I_r + SS^T)^{-1} R^{\dagger} \end{bmatrix}$$

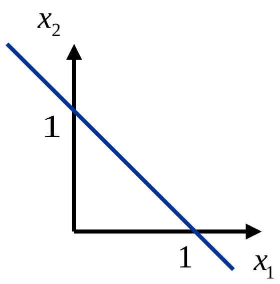
Copyright © 2008 by K. Pattipati

Why is this true? Look at LS Problem

- When rank(A) = $r < n \Rightarrow$ underdetermined system
- Less number of independent equations than unknowns.

$\Rightarrow$Infinite number of solutions satisfying $A\underline{x} = \underline{b}$

$$x_1 + x_2 = 1$$
$$4x_1 + 4x_2 = 4$$



$$\Rightarrow \quad J = \left(A\underline{x} - \underline{b}\right)^T \left(A\underline{x} - \underline{b}\right) = 0 \text{ for infinite number of } \underline{x}$$

- So, among these infinite number of $\{\underline{x}\}$, let us pick one that has minimum norm $\left(\underline{x}^T \underline{x}\right)^{1/2} \Rightarrow$ minimum Euclidean length.
- That is, solve

$$\min_{\underline{x}} \quad J_1 = \underline{x}^T \underline{x}$$
$$\text{s.t.} \quad AP\underline{x} = \underline{b}$$

 Copyright © 2008 by K. Pattipati

Optimality Conditions

- This is a **convex programming problem** and has a unique minimizing solution

$$\underline{x}^T\underline{x} = \underline{x}_1^T\underline{x}_1 + \underline{x}_2^T\underline{x}_2, \quad AP\underline{x} = \underline{b} \;\Rightarrow\; R\underline{x}_1 + RS\underline{x}_2 = \underline{b}$$

where $\underline{x}_1$ and $\underline{x}_2$ are of dimensions r and $(n-r)$, respectively.

- Define the Lagrangian function,

$$L = \underline{x}_1^T\underline{x}_1 + \underline{x}_2^T\underline{x}_2 + \underline{\lambda}^T\left[R\underline{x}_1 + RS\underline{x}_2 - \underline{b}\right]$$

- From Karush-Kuhn-Tucker's necessary and sufficient conditions of optimality for convex problems

$$\partial L/\partial\underline{x}_1 = 0 \Rightarrow 2\underline{x}_1 + R^T\underline{\lambda} = 0 \Rightarrow \underline{\lambda} = -2(R^T)^{-1}\underline{x}_1$$

$$\partial L/\partial\underline{x}_2 = 0 \Rightarrow 2\underline{x}_2 + S^TR^T\underline{\lambda} = 0$$

$$\partial L/\partial\underline{\lambda} = 0 \Rightarrow R\underline{x}_1 + RS\underline{x}_2 = \underline{b}$$

$$\underline{x}_2 = S^T\underline{x}_1 \quad \Rightarrow \underline{x}_1 = \left(I_r + SS^T\right)^{-1}R^{\dagger}\underline{b}$$

Thus, minimum norm satisfying $A\underline{x} = \underline{b}$ is given by:

$$\underline{x}_{LS} = \begin{bmatrix} \left(I_r + SS^T\right)^{-1}R^{\dagger} \\ S^T\left(I_r + SS^T\right)^{-1}R^{\dagger} \end{bmatrix}\underline{b} = (AP)^{\dagger}\underline{b}$$

Copyright © 2008 by K. Pattipati

Pseudo Inverse Mechanization - 1

- Suppose we have done the Gram-Schmidt procedure on *AP*

$$\text{i.e.,}\left[R \mid RS\right] = m\big[\overset{r}{Q} \mid \overset{n-r}{0}\big]U$$

$$\text{where} \qquad U = \begin{matrix} r \\ n-r \end{matrix}\begin{bmatrix} U_1 & W \\ 0 & I_{n-r} \end{bmatrix}$$

$$\text{Compute } U^{-1} = \begin{bmatrix} \overset{r}{U_1^{-1}} & \overset{n-r}{X} \\ 0 & I_{n-r} \end{bmatrix} \Rightarrow X = -U_1^{-1}W$$

$$\text{Since } (AP)U^{-1} = \left[Q \quad 0\right] \Rightarrow \left[R \quad RS\right]\begin{bmatrix} U_1^{-1} & X \\ 0 & I_{n-r} \end{bmatrix} = \left[RU_1^{-1} \quad RX + RS\right]$$

$$\Rightarrow R = QU_1 \qquad \Rightarrow R^{\dagger} = (U_1)^{-1}Q^T$$

$$\Rightarrow X = -S$$

$$\Rightarrow \text{last } (n-r) \text{ columns of } U^{-1} \text{ are } \begin{bmatrix} -S \\ I_{n-r} \end{bmatrix} \text{ at the point we hit dependency}$$

Apply G-S to these last $(n-r)$ columns

$$Z^T\left[-S^T \mid I_{n-r}\right] = \left[-Z^TS^T \mid Z^T\right]$$

Copyright © 2008 by K. Pattipati

Pseudo Inverse Mechanization - 2

- Note that the transformation automatically gets stored

$$\text{or} \begin{bmatrix} -S \\ I_{n-r} \end{bmatrix} Z = \begin{matrix} r \\ n-r \end{matrix} \begin{bmatrix} -SZ \\ Z \end{bmatrix}$$

$$\text{Since } (-Z^T S^T \;\; Z^T) \begin{bmatrix} -SZ \\ Z \end{bmatrix} = Z^T \left[I_{n-r} + S^T S \right] Z = I_{n-r} \text{ (Columns are orthogonal)}$$

$$\Rightarrow \quad I_{n-r} + S^T S = (Z^T)^{-1} Z^{-1} \Rightarrow \left(I_{n-r} + S^T S \right)^{-1} = Z Z^T$$

- Also, using Sherman-Morrison-Woodbury formula

$$\left(I_r + S S^T \right)^{-1} = I_r - S \left(I_{n-r} + S^T S \right)^{-1} S^T = I_r - (SZ)(SZ)^T$$

$$S^T \left(I_r + S S^T \right)^{-1} = S^T - S^T S.Z Z^T S^T = [I_{n-r} - S^T S \left(I_{n-r} + S^T S \right)^{-1}] S^T$$

$$= \left(S^T S + I_{n-r} \right)^{-1} S^T = Z Z^T S^T = Z(SZ)^T$$

$$\text{Recall} \quad (AP)^{\dagger} = \begin{bmatrix} \left[I_r - (SZ)(SZ)^T \right]^{-1} R^{\dagger} \\ Z(SZ)^T R^{\dagger} \end{bmatrix}$$

$$= \begin{matrix} r \\ n-r \end{matrix} \begin{bmatrix} U_1^{-1} & -SZ \\ 0 & Z \end{bmatrix} \begin{bmatrix} Q^T \\ (SZ)^T U_1^{-1} Q^T \end{bmatrix}$$

 Copyright © 2008 by K. Pattipati

Summary of Procedure

$$\begin{bmatrix} AP \\ I_n \end{bmatrix} \to \begin{bmatrix} R & S \\ I_r & 0 \\ 0 & I_{n-r} \end{bmatrix} \xrightarrow{\text{GS}} \begin{bmatrix} Q & 0 \\ U_1^{-1} & -S \\ 0 & I \end{bmatrix} \xrightarrow{\text{GS}\begin{bmatrix} -S \\ I \end{bmatrix}} \begin{bmatrix} Q & 0 \\ U_1^{-1} & -SZ \\ 0 & Z \end{bmatrix}$$

$$\begin{bmatrix} Q & 0 \\ U_1^{-1} & -SZ \\ (SZ)^T U_1^{-1} & Z \end{bmatrix} \to \begin{bmatrix} Q & \left[(SZ)^T U_1^{-1} Q^T \right]^T \\ U_1^{-1} & -SZ \\ 0 & Z \end{bmatrix}$$

Finally, $\begin{bmatrix} U_1^{-1} & -SZ \\ 0 & Z \end{bmatrix} \begin{bmatrix} Q^T \\ (SZ)^T U_1^{-1} Q^T \end{bmatrix} = (AP)^\dagger$

Note that we can store $A^\dagger$ in A,i.e., in the space occupied by

$$\left[Q \mid \ ((SZ)^T U_1^{-1} Q^T)^T \right]$$

Finally, we can permute rows to obtain $A^\dagger$

$$A^\dagger = P(AP)^\dagger = P_1 P_2 \ldots P_r (AP)^\dagger$$

$\Rightarrow$ swap row r with c_r,....,row 1 with c_1

Copyright © 2008 by K. Pattipati

Householder Method to compute A† - 1

- Householder on *AP* gives

$$Q^T AP = \begin{array}{c} r \\ m-r \end{array} \overset{\begin{array}{cc} r & n\text{-}r \end{array}}{\begin{bmatrix} U_1 & W \\ 0 & 0 \end{bmatrix}} \quad r = \text{rank(A)}$$

- Zero out *W* via Householder again

Consider $Z_r \ldots Z_1 \begin{array}{c} r \\ n-r \end{array} \begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} \tilde{U}_1^T \\ 0 \end{bmatrix} \begin{array}{c} r \\ n-r \end{array}$ can do it in $r^2(n-r)$ flops.

$$Q^T APZ_1 \ldots Z_r = \begin{bmatrix} \tilde{U}_1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$Q^T (AP)Z = \begin{bmatrix} \tilde{U}_1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{where } Z = Z_1 Z_2 \ldots Z_r$$

$$\Rightarrow AP = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} \tilde{U}_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}^T$$

$$(AP)^\dagger = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \begin{bmatrix} \tilde{U}_1^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}^T$$

$$= \begin{bmatrix} Z_{11}\tilde{U}_1^{-1}Q_{11}^T & Z_{11}\tilde{U}_1^{-1}Q_{21}^T \\ Z_{21}\tilde{U}_1^{-1}Q_{11}^T & Z_{21}\tilde{U}_1^{-1}Q_{22}^T \end{bmatrix}$$

Copyright © 2008 by K. Pattipati

Householder Method to compute $A^\dagger$ - 2

- Swap rows to get $A^\dagger$
- Can solve LS problem easily

$$\|A\underline{x}-\underline{b}\|_2^2 = \|(Q^T APZZ^T P^T \underline{x} - Q^T \underline{b}\|_2^2$$

$$= \|\tilde{U}_1 \underline{w} - \underline{c}\|_2^2 + \|\underline{d}\|_2^2 \;;\; Z^T P^T \underline{x} = \begin{bmatrix} \underline{w}^T \underline{y}^T \end{bmatrix}^T \;;\; Q^T \underline{b} = \begin{bmatrix} \underline{c}^T \underline{d}^T \end{bmatrix}^T$$

$$\Rightarrow \text{ Solve } \tilde{U}_1 \underline{w} = \underline{c} \qquad \Rightarrow \underline{x} = PZ \begin{bmatrix} w \\ \underline{0} \end{bmatrix}$$

$$\text{Set } \underline{y} = 0 \qquad = (PZ)_r w$$

$\Rightarrow \underline{x}_{\text{LS}}$=weighted sum of first r columns of PZ

- Example:

$$A = \begin{bmatrix} 4 & 2 & 3 \\ 0 & 1 & 5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} 8 \\ 2 \\ 5 \\ 7 \end{bmatrix}$$

A is already in the form $\begin{bmatrix} U_1 & W \\ 0 & 0 \end{bmatrix}$ with $U_1 = \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix}$ and $W = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$

Copyright © 2008 by K. Pattipati

Example via Householder

Consider $\begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 2 & 1 \\ 3 & 5 \end{bmatrix}$

$$\underline{u}_1 = \begin{bmatrix} 4+\sqrt{29} \\ 2 \\ 3 \end{bmatrix};\ Z_1 = \left[I - \frac{2\underline{u}_1\underline{u}_1^T}{\underline{u}_1\underline{u}_1^T} \right] = \begin{bmatrix} -0.7428 & -0.3714 & -0.5571 \\ -0.3714 & 0.9209 & -0.1187 \\ -0.5571 & -0.1187 & 0.8219 \end{bmatrix}$$

$$Z_1 \begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} -5.3852 & -3.1568 \\ 0 & 0.3273 \\ 0 & 3.9909 \end{bmatrix}$$

$$Z_2 Z_1 \begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} -5.3852 & -3.1568 \\ 0 & -4.0043 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{U}_1^T \\ 0 \end{bmatrix}$$

$$\underline{w} = \tilde{U}_1^{-1} \begin{bmatrix} 8 \\ 2 \end{bmatrix} = \begin{bmatrix} -1.4856 \\ 0.6717 \end{bmatrix}$$

$$\underline{x}_{\text{LS}} = Z_1 Z_2 \begin{bmatrix} \underline{w} \\ 0 \end{bmatrix} = \begin{bmatrix} 1.4968 \\ 0.5806 \\ 0.2839 \end{bmatrix}$$

Copyright © 2008 by K. Pattipati

$A^\dagger$ Via SVD: Best Method - 1

- Best method yet to come……Lecture 12
- Reduce A to upper Δ form via Householder

$$Q_R^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

- Reduce R to bi-diagonal form via Householder

$$Q_B^T R S_B = B_1 = \begin{bmatrix} d_1 & f_1 & 0 & 0 & .. & 0 \\ 0 & d_2 & f_2 & 0 & .. & 0 \\ .. & .. & .. & .. & .. & .. \\ 0 & .. & .. & .. & d_{n-1} & f_n \\ 0 & .. & .. & .. & .. & d_n \end{bmatrix}$$

- Zero the super-diagonal elements via a symmetric QR algorithm for Eigen values (Lecture 11)

$$Q_\Sigma^T B S_\Sigma = \Sigma = diag(\sigma_1 \sigma_2 ... \sigma_n), \text{ and } \quad B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$$

$$Q_\Sigma^T (Q_B^T I_{m-n}) Q_R^T A S_B S_\Sigma = \Sigma \; ; \; U = Q_R (Q_B I_{m-n}) Q_\Sigma \; ; \; V = S_B S_\Sigma$$

$$A = U \Sigma V^T \Rightarrow A^\dagger = V \Sigma^\dagger U^T$$

$$A^\dagger = \sum_{i=1}^{n} \frac{\underline{v}_i \underline{u}_i^T}{\sigma_i}$$

Copyright © 2008 by K. Pattipati

Iterative Improvement of Inverse

- Given $X_0 = A_0 \sim A^{-1}$, find X_1 better than X_0
- The method is based on Newton's method for solving $f(x) = 0$

  $=> \; x_{n+1} = x_n - f(x_n)/f'(x_n)$
- Applying the formula to $f(x) = a - 1/x$ (scalar) to get

$$x_{n+1} = x_n - \left[a - 1/x_n\right] / \left[1/x_n^2\right] = x_n + x_n\left(1 - a x_n\right) = x_n + \left(1 - x_n a\right) x_n$$

So, $x_{n+1} = x_n + e_n x_n$; e_n=error at iteration n.

Extending to matrices

$$X_{n+1} = X_n + \left(I - X_n A\right) X_n = E_n X_n \;\; ; \;\; X_0 = \text{intial estimate}$$

$$E_{n+1} = I - X_{n+1} A = I - X_n A - X_n A + (X_n A)^2 = (I - X_n A)^2$$

$$E_{n+1} = I - X_{n+1} A; \quad E_n = I - X_n A$$

$$\Rightarrow E_{n+1} = E_n^2$$

$\Rightarrow$ Rapid convergence provided $\left\| I - X_0 A \right\| < 1.$

Quadratic convergence

- Typically requires $2n^3$ MADDS / iteration (expensive)
- The procedure is valid for $\boldsymbol{A}^{\dagger}$ as well

Copyright © 2008 by K. Pattipati

Summary

- Givens Transformations
  - local effect
  - parallelization

- Weighted Least Squares Problem and its Solutions via Householder Transformation

- Computation of Pseudo (Generalized) Inverse
  - Gram-Schmidt
  - Householder
  - SVD

Copyright © 2008 by K. Pattipati