$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}_{m\times n}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1}}_{b}$$

CX1104: **Linear Algebra for Computing**

Chap. No : **8.4.4**

Lecture : **Eigen and Singular Values**

Topic : **SVD & Pseudoinverse**

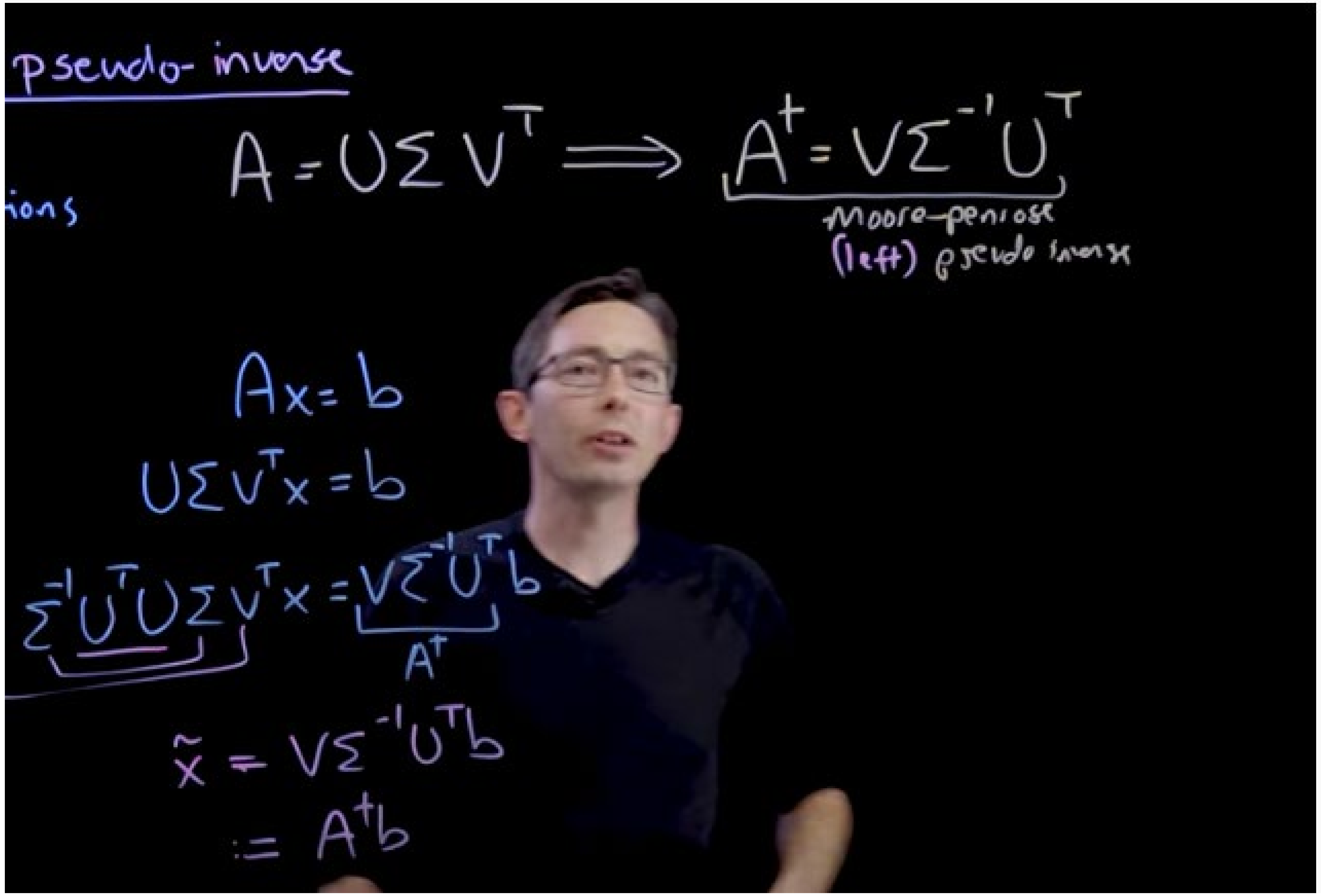
Concept : **Pseudoinverse & Least Squares Solution**

Instructor: **A/P Chng Eng Siong**

TAs: **Zhang Su, Vishal Choudhari**

Rev: 27th July 2020

Reference: Pseudoinverse by Steve Brunton and Matlab



Moore-Penrose Pseudoinverse

The Moore-Penrose pseudoinverse is a matrix that can act as a partial replacement for the matrix inverse in cases where it does not exist. This matrix is frequently used to solve a system of linear equations when the system does not have a unique solution or has many solutions.

For any matrix `A`, the pseudoinverse `B` exists, is unique, and has the same dimensions as `A'`. If `A` is square and not singular, then `pinv(A)` is simply an expensive way to compute `inv(A)`. However, if `A` is not square, or is square and singular, then `inv(A)` does not exist. In these cases, `pinv(A)` has some (but not all) of the properties of `inv(A)`:

Matrix A need not be square!

pinv

R2018b

Moore-Penrose pseudoinverse

collapse all in page

Syntax

```
B = pinv(A)
B = pinv(A,tol)
```

Description

example

`B = pinv(A)` returns the Moore-Penrose Pseudoinverse of matrix `A`.

`B = pinv(A,tol)` specifies a value for the tolerance. `pinv` treats singular values of `A` that are smaller than the tolerance as zero.

Ref: https://www.youtube.com/watch?v=PjeOmOz9jSY

Pseudoinverse Definition

Definition [edit]

For $A \in \mathbb{K}^{m\times n}$, a pseudoinverse of A is defined as a matrix $A^+ \in \mathbb{K}^{n\times m}$ satisfying all of the following four criteria, known as the Moore–Penrose conditions:[7][8]

1. $AA^+A = A$ (AA^+ need not be the general identity matrix, but it maps all column vectors of A to themselves);
2. $A^+AA^+ = A^+$ (A^+ acts like a weak inverse);
3. $(AA^+)^* = AA^+$ (AA^+ is Hermitian);
4. $(A^+A)^* = A^+A$ (A^+A is also Hermitian).

A^+ exists for any matrix A, but, when the latter has full rank (that is, the rank of A is $\min\{m,n\}$), then A^+ can be expressed as a simple algebraic formula.

In particular, when A has linearly independent columns (and thus matrix A^*A is invertible), A^+ can be computed as

$$A^+ = (A^*A)^{-1}A^*.$$

This particular pseudoinverse constitutes a *left inverse*, since, in this case, $A^+A = I$.

When A has linearly independent rows (matrix AA^* is invertible), A^+ can be computed as

$$A^+ = A^*(AA^*)^{-1}.$$

This is a *right inverse*, as $AA^+ = I$.

Left inverse: Multiplying A with A^+ from the left yields I.
Right inverse: Multiplying A with A^+ from the right yields I.

$$\boxed{A \text{ Hermitian} \iff a_{ij} = \overline{a_{ji}}}$$

or in matrix form:

$$A \text{ Hermitian} \iff A = \overline{A^{\mathsf{T}}}.$$

Ref:
https://en.wikipedia.org/wiki/Moore–Penrose_inverse
https://en.wikipedia.org/wiki/Hermitian_matrix

Reduced SVD & Pseudoinverse

EXAMPLE 7 (Reduced SVD and the Pseudoinverse of A) When Σ contains rows or columns of zeros, a more compact decomposition of A is possible. Using the notation established above, let $r = \text{rank } A$, and partition U and V into submatrices whose first blocks contain r columns:

$$U = [\, U_r \quad U_{m-r} \,], \quad \text{where } U_r = [\, \mathbf{u}_1 \quad \cdots \quad \mathbf{u}_r \,]$$
$$V = [\, V_r \quad V_{n-r} \,], \quad \text{where } V_r = [\, \mathbf{v}_1 \quad \cdots \quad \mathbf{v}_r \,]$$

Then U_r is $m \times r$ and V_r is $n \times r$. (To simplify notation, we consider U_{m-r} or V_{n-r} even though one of them may have no columns.) Then partitioned matrix multiplication shows that

$$A = [\, U_r \quad U_{m-r} \,] \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{n-r}^T \end{bmatrix} = U_r D V_r^T \tag{9}$$

This factorization of A is called a **reduced singular value decomposition** of A. Since the diagonal entries in D are nonzero, D is invertible. The following matrix is called the **pseudoinverse** (also, the **Moore–Penrose inverse**) of A:

$$A^+ = V_r D^{-1} U_r^T \tag{10}$$

Supplementary Exercises 12–14 at the end of the chapter explore some of the properties of the reduced singular value decomposition and the pseudoinverse. ■

Lay, Linear Algebra and its Applications (4th Edition)

Pseudo Inverse Example:

```
============\n test_SVD 8.4.4 \n ---

A =

     1     1
     1     2
     1     3

rank A =

ans =

     2

U =

   -0.3231    0.8538    0.4082
   -0.5475    0.1832   -0.8165
   -0.7719   -0.4873    0.4082

D =

    4.0791         0
         0    0.6005
         0         0

V =

   -0.4027    0.9153
   -0.9153   -0.4027
```

```
disp("Reconstruct A from : U*D*V'");
A_recon1 = U*D*V'
```

```
A_recon1 =

    1.0000    1.0000
    1.0000    2.0000
    1.0000    3.0000
```

$$A = \widetilde{U_r}\widetilde{D_r}\widetilde{V_r}^T$$ // using r=2

```
r=2;
Ur = U(:,1:r);
Dr = D(1:r,1:r)
Dr_inv = diag(diag(Dr).^(-1));
Vr = V(:,1:r);
disp('Reconstruct A from r=2 singularValues/Vectors ');
A_recon = Ur*Dr*Vr'
```

```
Reconstruct A from r=2 singularValues/Vectors

A_recon =

    1.0000    1.0000
    1.0000    2.0000
    1.0000    3.0000
```

```
disp("generated pInv using SVD matrixes:");
A_pInv  = Vr*Dr_inv*Ur'
disp("checking using pinv(A)"); pinv(A)
disp('showing pinv(A)*A = I');
disp('// bcos full col rank (left inverse exist)')
pinv(A)*A
```

```
generated pInv using SVD matrixes:

A_pInv =

    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000

checking using pinv(A)

ans =

    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000

showing pinv(A)*A = I
// bcos full col rank (left inverse exist)

ans =

    1.0000   -0.0000
    0.0000    1.0000
```

$$A^+ = \widetilde{V_r}\widetilde{D_r^{-1}}\widetilde{U_r}^T$$
// using r=2

Pseudo Inverse Example: Properties of A^+

1. $AA^+A = A$

```
disp("Property1: (A*pinv(A))*A == A")
(A*pinv(A))*A
disp("projectionColSpaceOfA")
ProjectionColA= (A*pinv(A))
ProjectionColA*ProjectionColA
```

```
Property1: (A*pinv(A))*A == A

ans =

    1.0000    1.0000
    1.0000    2.0000
    1.0000    3.0000

projectionColSpaceOfA

ProjectionColA =

    0.8333    0.3333   -0.1667
    0.3333    0.3333    0.3333
   -0.1667    0.3333    0.8333


ans =

    0.8333    0.3333   -0.1667
    0.3333    0.3333    0.3333
   -0.1667    0.3333    0.8333
```

2. $A^+AA^+ = A^+$

```
disp("Property2: pinv(A)*A*pinv(A) == pinv(A)")
P1=pinv(A)*A*pinv(A)
P2=pinv(A)
P3=pinv(A)*A
```

```
P1 =

    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000


P2 =

    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000


P3 =

    1.0000   -0.0000
    0.0000    1.0000
```

Pseudoinverse and Least Squares Solution

EXAMPLE 8 (Least-Squares Solution) Given the equation $A\mathbf{x} = \mathbf{b}$, use the pseudoinverse of A in (10) to define

$$\hat{\mathbf{x}} = A^{+}\mathbf{b} = V_r D^{-1} U_r^T \mathbf{b}$$

Then, from the SVD in (9),

$$\begin{aligned} A\hat{\mathbf{x}} &= (U_r D V_r^T)(V_r D^{-1} U_r^T \mathbf{b}) \\ &= U_r D D^{-1} U_r^T \mathbf{b} \qquad \text{Because } V_r^T V_r = I_r \\ &= U_r U_r^T \mathbf{b} \end{aligned}$$

It follows from (5) that $U_r U_r^T \mathbf{b}$ is the orthogonal projection $\hat{\mathbf{b}}$ of $\mathbf{b}$ onto Col A. (See Theorem 10 in Section 6.3.) Thus $\hat{\mathbf{x}}$ is a least-squares solution of $A\mathbf{x} = \mathbf{b}$. In fact, this $\hat{\mathbf{x}}$ has the smallest length among all least-squares solutions of $A\mathbf{x} = \mathbf{b}$. See Supplementary Exercise 14. ■

```
disp('least squares solution using SVD');
A
b = [1 1 10]'
pinv_A = pinv(A)
disp('pinv_A*A');  pinv_A*A
disp('hat_x = pinv(A)*b:  ');
hat_x = pinv_A*b

est_b = A*hat_x
```

```
A =

     1     1
     1     2
     1     3


b =

     1
     1
    10

pinv_A =

    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000

pinv_A*A

ans =

    1.0000   -0.0000
    0.0000    1.0000

hat_x = pinv(A)*b:

hat_x =

   -5.0000
    4.5000

est_b =

   -0.5000
    4.0000
    8.5000
```

Lay, Linear Algebra and its Applications (4th Edition)

422 CHAPTER 7

LS solution Is same as pinv solution

```
disp('Using the LS solution:');
hat_x2 = inv(A'*A)*A'*b
```

```
hat_x2 =

   -5.0000
    4.5000
```

The above is possible using the LS solution because
Inv(A’*A) exist. This is because there is full column rank == 2.

Pseudoinverse and Least Squares Solution

Review:

EXAMPLE 6 (Bases for Fundamental Subspaces) Given an SVD for an $m \times n$ matrix A, let $\mathbf{u}_1, \dots, \mathbf{u}_m$ be the left singular vectors, $\mathbf{v}_1, \dots, \mathbf{v}_n$ the right singular vectors, and $\sigma_1, \dots, \sigma_n$ the singular values, and let r be the rank of A. By Theorem 9,

$$\{\mathbf{u}_1, \dots, \mathbf{u}_r\} \tag{5}$$

is an orthonormal basis for Col A.

THEOREM 10 If $\{\mathbf{u}_1, \dots, \mathbf{u}_p\}$ is an orthonormal basis for a subspace W of $\mathbb{R}^n$, then

$$\text{proj}_W\,\mathbf{y} = (\mathbf{y}\cdot\mathbf{u}_1)\mathbf{u}_1 + (\mathbf{y}\cdot\mathbf{u}_2)\mathbf{u}_2 + \cdots + (\mathbf{y}\cdot\mathbf{u}_p)\mathbf{u}_p \tag{4}$$

If $U = [\,\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_p\,]$, then

$$\text{proj}_W\,\mathbf{y} = UU^T\mathbf{y} \quad \text{for all } \mathbf{y} \text{ in } \mathbb{R}^n \tag{5}$$

PROOF Formula (4) follows immediately from (2) in Theorem 8. Also, (4) shows that $\text{proj}_W\,\mathbf{y}$ is a linear combination of the columns of U using the weights $\mathbf{y}\cdot\mathbf{u}_1$, $\mathbf{y}\cdot\mathbf{u}_2, \dots, \mathbf{y}\cdot\mathbf{u}_p$. The weights can be written as $\mathbf{u}_1^T\mathbf{y}, \mathbf{u}_2^T\mathbf{y}, \dots, \mathbf{u}_p^T\mathbf{y}$, showing that they are the entries in $U^T\mathbf{y}$ and justifying (5). ■

Lay, Linear Algebra and its Applications (4th Edition)