

CSE276C - Linear Systems of Equations

Henrik I. Christensen



Computer Science and Engineering
University of California, San Diego
<http://cri.ucsd.edu>

September 2021

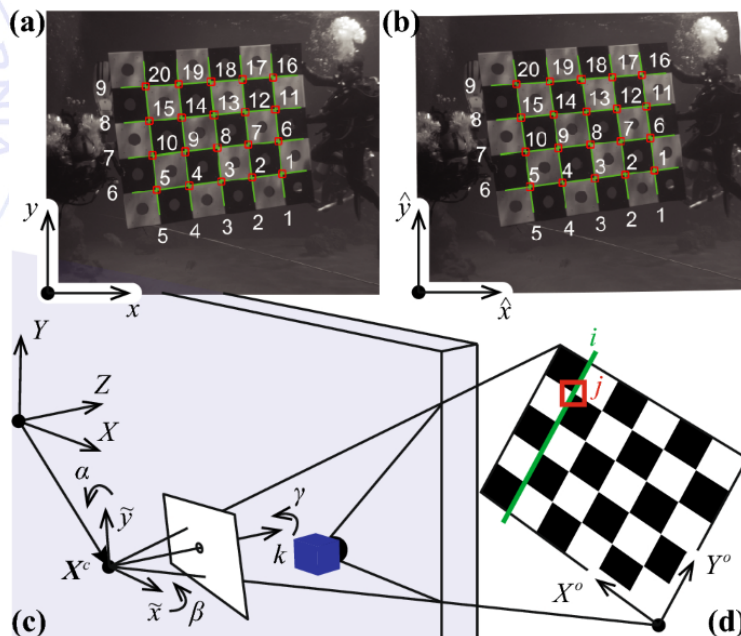
Outline

- Linear Systems of Equations
- Solution Techniques - Gauss Jordan
- Matrix Decomposition
- Matrix Factorization
- Singular Value Decomposition
- Rank and sensitivity

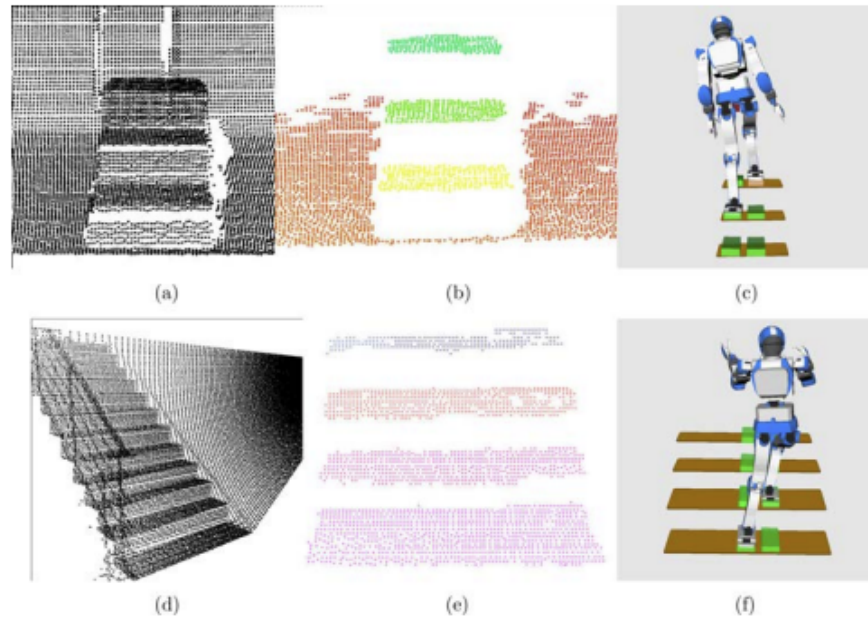
Material

- Numerical Recipes: Chapter 2
- Math for ML: Chapter 2.1-2.3

Example: Camera calibration



Example: Plane Estimation



Linear Systems of Equations

- One of the most basic tasks is solve for a set of unknowns

$$\begin{aligned} a_{00}x_0 + a_{01}x_1 + a_{02}x_2 + \dots + a_{0n-1}x_{n-1} &= b_0 \\ a_{10}x_0 + a_{11}x_1 + a_{12}x_2 + \dots + a_{1n-1}x_{n-1} &= b_1 \\ &\vdots \\ a_{m-10}x_0 + a_{m-11}x_1 + a_{m-12}x_2 + \dots + a_{m-1n-1}x_{n-1} &= b_{m-1} \end{aligned}$$

Linear Systems of Equations

- One of the most basic tasks is solve for a set of unknowns

$$\begin{aligned} a_{00}x_0 + a_{01}x_1 + a_{02}x_2 + \dots + a_{0n-1}x_{n-1} &= b_0 \\ a_{10}x_0 + a_{11}x_1 + a_{12}x_2 + \dots + a_{1n-1}x_{n-1} &= b_1 \\ &\vdots \\ a_{m-10}x_0 + a_{m-11}x_1 + a_{m-12}x_2 + \dots + a_{m-1n-1}x_{n-1} &= b_{m-1} \end{aligned}$$

- which we can rewrite

$$\mathbf{A}\vec{x} = \vec{b}$$

where

$$\mathbf{A} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & \dots & a_{0n-1} \\ a_{10} & a_{11} & a_{12} & \dots & a_{1n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m-10} & a_{m-11} & a_{m-12} & \dots & a_{m-1n-1} \end{pmatrix}, \vec{b} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{m-1} \end{pmatrix}$$

Matrix Properties

- Given an $m \times n$ matrix A we define
 - Column space - Linear combination of columns
 - Row space - Linear combination of row
- We can consider A a mapping:

$$A : R^n \rightarrow R^m$$
$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} \rightarrow \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{pmatrix} = \mathbf{A} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

- Column space of A is vector subspace of R^m that image vectors under A

Null Space

- We define the null-space: set of vectors $x \in R^n$ where

$$Ax = 0$$

- The row space and the null space are complementary

$$n = \dim(\text{row space}) + \dim(\text{null space})$$

Questions

Questions

Matrix properties

- Consider the square matrix A . The square matrix B is the inverse if

$$AB = I_n = BA$$

and we denote this A^{-1} .

- If the inverse exists the matrix is called regular/invertible/non-singular
- Inverse matrices are unique
- If the determinant of A : $\det(A)$ is zero the matrix is singular
- The transpose of A is denoted A^T and elements of the transpose are $a_{ji}^T = a_{ij}$
- useful properties

$$\begin{aligned} AA^{-1} &= I = A^{-1}A \\ (AB)^{-1} &= B^{-1}A^{-1} \\ (A+B)^{-1} &\neq A^{-1} + B^{-1} \\ (A^T)^T &= A \\ (A+B)^T &= A^T + B^T \\ (AB)^T &= B^T A^T \end{aligned}$$

Singular matrices

- A matrix **A** is **singular** iff
 - $\det(A) = 0$
 - $\text{rank}(A) < n$
 - rows of A are not linearly independent
 - columns of A are not linearly independent
 - the dimension of the null-space of A is non-zero
 - A is not invertible

Gauss-Jordan Elimination

- How can we solve the equation system?
- The standard form

$$\mathbf{A}\vec{x} = \vec{b} \rightarrow \mathbf{U}\vec{x}' = \vec{b}'$$

where

$$\mathbf{U} = \begin{pmatrix} d_0 & & U'_m \\ & \ddots & \\ 0 & & d_{n-1} \end{pmatrix}$$

Gauss-Jordan Elimination

- How can we solve the equation system?
- The standard form

$$\mathbf{A}\vec{x} = \vec{b} \rightarrow \mathbf{U}\vec{x}' = \vec{b}'$$

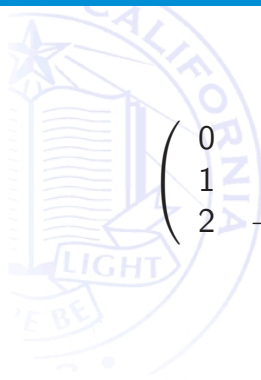
where

$$\mathbf{U} = \begin{pmatrix} d_0 & & U'_m \\ & \ddots & \\ 0 & & d_{n-1} \end{pmatrix}$$

- Two different approaches:
 - 1 Gauss Elimination - $Ux' = b'$
 - 2 Gauss Jordan - $Dx^* = b^*$

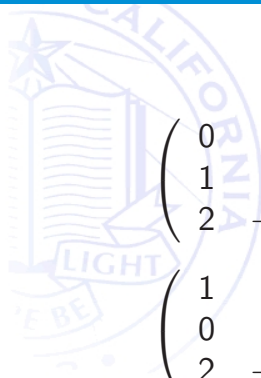
Allows for direct back substitution

Example of Elimination



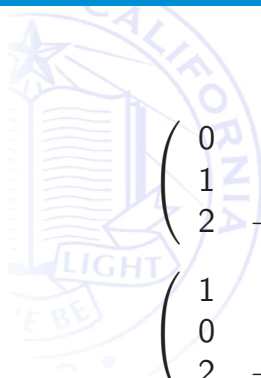
$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 4 & -1 & | & 5 \\ 1 & 1 & 1 & | & 6 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$

Example of Elimination



$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 4 & -1 & | & 5 \\ 1 & 1 & 1 & | & 6 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & | & 6 \\ 0 & 4 & -1 & | & 5 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$

Example of Elimination

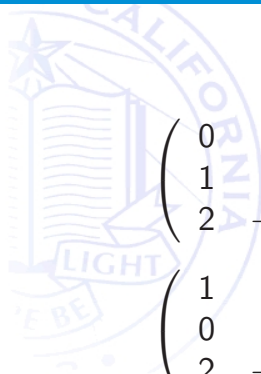


$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 4 & -1 & | & 5 \\ 1 & 1 & 1 & | & 6 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & | & 6 \\ 0 & 4 & -1 & | & 5 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & -4 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -11 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & | & 6 \\ 0 & 4 & -1 & | & 5 \\ 0 & -4 & -1 & | & -11 \end{pmatrix}$$

Example of Elimination



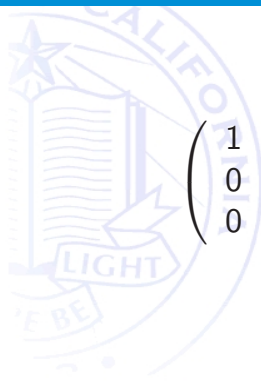
$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 4 & -1 & | & 5 \\ 1 & 1 & 1 & | & 6 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & | & 6 \\ 0 & 4 & -1 & | & 5 \\ 2 & -2 & 1 & | & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & -4 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -11 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & | & 6 \\ 0 & 4 & -1 & | & 5 \\ 0 & -4 & -1 & | & -11 \end{pmatrix}$$

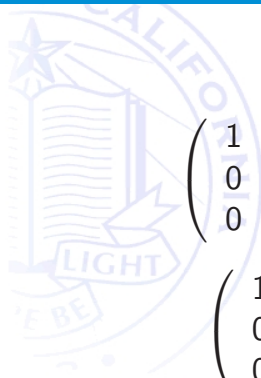
$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -6 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 1 & | & 6 \\ 0 & 4 & -1 & | & 5 \\ 0 & 0 & -2 & | & -6 \end{pmatrix}$$

Gauss Elimination → Gauss Jordan



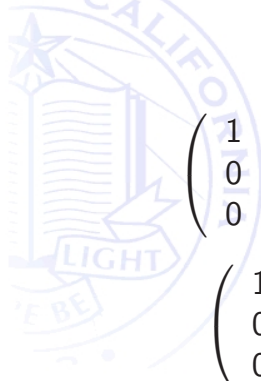
$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

Gauss Elimination → Gauss Jordan



$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & 0 & 1 & 3 \end{array} \right)$$
$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & 0 & 8 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

Gauss Elimination → Gauss Jordan

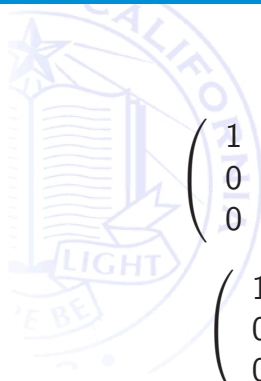


$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & 0 & 8 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

Gauss Elimination → Gauss Jordan



$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & 0 & 8 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

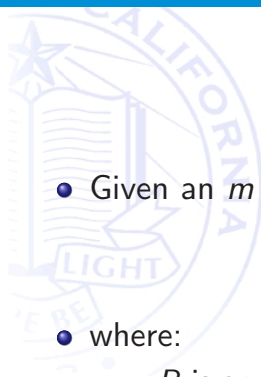
$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right)$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array} \right)$$



Questions

Matrix Decomposition



- Given an $m \times n$ matrix we can write \mathbf{A} in the form

$$\mathbf{PA} = \mathbf{LDU}$$

- where:
 - P is an $m \times m$ permutation matrix that specs row interchanges
 - L is a lower triangular matrix with 1 along the diagonal
 - U is a upper triangular matrix with 1 along the diagonal
 - D is a square diagonal only matrix
- If \mathbf{A} is a symmetric positive definite then $\mathbf{U} = \mathbf{L}^T$ and D has strictly positive diagonal elements

Solving the matrix system

- Our objective is to solve

$$\begin{aligned} LDUx &= Pb && \text{which we can solve} \\ Ly &= Pb && \text{(solve for } y) \\ Ux &= D^{-1}y && \text{(solve for } x) \end{aligned}$$

- Enable use of forward / backward substitution

Square - Full Rank Matrices

- If \mathbf{A} is a square $n \times n$ matrix with n linearly independent eigen vectors, then

$$\mathbf{A} = \mathbf{S}\mathbf{E}\mathbf{S}^{-1}$$

where

- \mathbf{E} is a diagonal matrix where elements are the eigenvalues of \mathbf{A}
- \mathbf{S} is a matrix where the columns are the eigenvectors of \mathbf{A}
- Any solution is then a linear combination of basis vectors. Useful for example for sub-space methods (discussed later)

Matrix factorization based on $A^T A$

- We will look at QR and SVD decompositions in more detail
- Consider A has independent columns then we can factorize

$$A = QR$$

where Q is $m \times n$ and R is $n \times n$

- Q has the same column space as A but it is orthonormal, i.e., $Q^T Q = I$
- R is upper triangular
- Two possible approaches:
 - Use Gram Schmidt to orthogonalize A . The columns are now an orthonormal basis, R is computed by keep track of the G-S operations. R expresses the linear combinations of Q to form A .
 - i) Form $A^T A$, ii) compute LDU factorization, iii) $R = D^{\frac{1}{2}} L^T$ and $Q = AR^{-1}$
- More efficient QR factorizations exist (see Numerical Recipes) in general $O(n^3)$

Gram-Schmidt?

- Build an orthonormal basis by re-projection
- Build a basis using $proj_u(v) = \frac{\langle v, u \rangle}{\langle u, u \rangle} u$, i.e., project v onto u
- Process is then
 - $u_1 = v_1$
 - $u_2 = v_2 - proj_{u_1}(v_2)$
 - $u_3 = v_3 - proj_{u_1}(v_3) - proj_{u_2}(v_3)$
 - $u_k = v_k - \sum_{j=1}^{k-1} proj_{u_j}(v_k)$
 - $e_i = \frac{u_i}{||u_i||}$ as the normal basis vectors

Applications

- QR: is an iterative process of building a factorization / eigenvectors
- If we wish to solve a system $Ax = b$ in the LSQ sense

$$\bar{x} = (A^T A)^{-1} A^T b$$

given full rank $Q^T Q = I$ i.e. with a QR factorization

$$\bar{x} = R^{-1} Q^T b$$

compute $Q^T R$ and back substitute for $R\bar{x} = Q^T b$ more stable than $A^T A\bar{x} = A^T b$, i.e., the Moore-Penrose pseudo inverse

Questions

Questions

Singular Value Decomposition

- We can factorize any $m \times n$ matrix A as

$$A = UDV^T$$

where

- U is an $m \times m$ w. columns are the eigenvectors of $A^T A$
- D is a diagonal matrix

$$D = \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_k & & \\ & & & 0 & \\ & 0 & & 0 & \\ & & & & 0 \end{pmatrix}$$

where $\sigma_1 > \dots > \sigma_k > 0$ and the $\text{rank}(A) = k$

- σ_i are sqrt of eigenvalues of $A^T A$ and called the singular values
- if A is symmetric and positive definite then $U = V^T$ and D is the eigenvalue matrix of A

Question

You are telling us all this why?

Motivation

- Goal is to solve

$$Ax = b$$

- For all A and b
- In a numerically stable manner
- Solve equation in reasonable time
- Comments
 - Ideally we would like for an $n \times n$ matrix

$$x = A^{-1}b$$

- If A is under-constrained the full solution set
- If A is over-constrained the LSQ solution

Considerations

- 1 Gauss Elimination is efficient, but necessarily stable

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1.01 & 1.00 & 1.00 \\ 1.00 & 1.01 & 1.00 \\ 1.00 & 1.00 & 1.01 \end{pmatrix}$$

Independent *Independent?*

not well suited for close to singular or over-constrained systems

- 2 Can we do elimination and solve

$$Ly = b \text{ and } Ux = D^{-1}y$$

if A is close to singular D^{-1} could be a challenge

Eigenvector factorization

- Remembers we can factorize a square matrix

$$A = SES^{-1}$$

where E is the eigenvalue matrix and S is the eigenvector matrix

- We can add this to the trick of working with $A^T A$ or AA^T
- We can use

$$A^T A = VDV^T$$

and

$$AA^T = UD'U^T$$

- Where D is the eigenvalue of $A^T A$, V are the eigenvector of $A^T A$, D' are the eigenvalue of AA^T and U are eigenvectors of AA^T
- We can decompose

$$A = UDV^T$$

- Note:
 - $\text{rank}(A) = \text{rank}(D) = k$
 - $\text{colspace}(A) = \text{first } k \text{ columns of } U$
 - $\text{nullspace}(A) = \text{first } n-k \text{ columns of } V$

Numerical considerations

- If SVD generates ≈ 0 eigenvalues the best is zero them out (compare values, see later)
- Example we had before

$$\begin{pmatrix} 1.01 & 1.00 & 1.00 \\ 1.00 & 1.01 & 1.00 \\ 1.00 & 1.00 & 1.01 \end{pmatrix}$$

the D matrix is then

$$\begin{pmatrix} 3.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

so you barely have full rank.

Sensitivity

- If we use

$$A = UDV^T \text{ then using } \sum_{i=1}^n \sigma_i u_i v_j$$

solving for $Ax = b$ is then

$$x = A^{-1}b = (UDV^T)^{-1}v \Rightarrow \sum \frac{u_i b}{\sigma_i} v_j$$

as σ_i decreases we have a sensitivity problem

- The condition number is a good indicator

$$K(A) = \frac{\sigma_1}{\sigma_k}$$

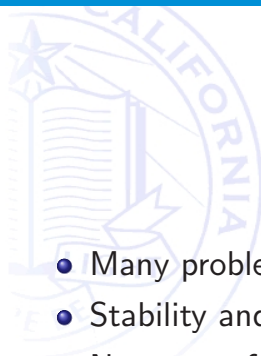
Using SVD

- To solve $Ax = b$ we can compute

$$\bar{x} = V \frac{1}{D} U^T b$$

- The solution is
 - If A is non-singular then \bar{x} is the unique solution
 - If A is singular then \bar{x} is the solution is closest to origin when b is range
 - If A is singular and b is not in range then \bar{x} is the LSQ solution
- You can use SVD for all your needs to solve the equations $Ax = b$

Linear Systems of Equations



- Many problems in robotics can be solved using linear systems of equations
- Stability and sensitivity are key to consider
- Numerous factorization methods available - QR and SVD merely two of them
- You can use numerous tricks to make problems tractable
- Factorization part of all the big packages - NumPy, Matlab, Linpack, ...

Questions



Questions