CSE276C - Linear Systems of Equations

Henrik I. Christensen

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

October 2020

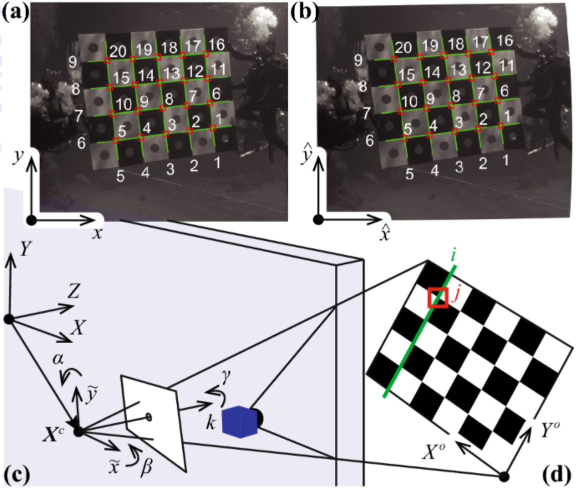
Outline

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

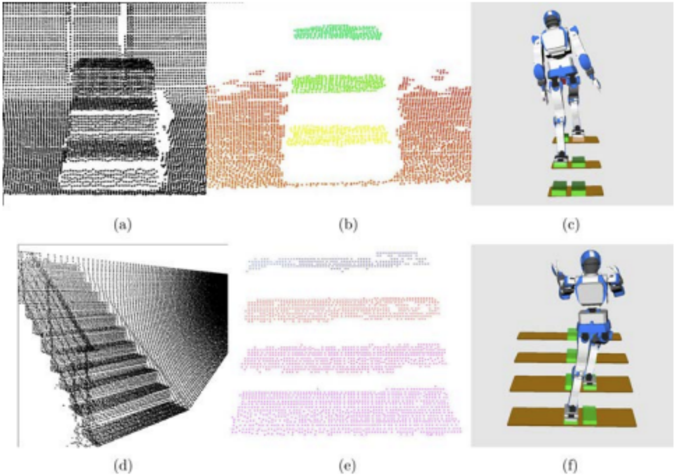
Material

- 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

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

Linear Systems of Equations

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

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

- which we can rewrite

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

where

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

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:

$$A : R^n \rightarrow R^m$$

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

- 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

$$n = dim(row\ space) + dim(null\ space)$$

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/invertable/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

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

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

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

where

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

Gauss-Jordan Elimination

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

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

where

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

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

  Allows for direct back substitution

Example of Elimination

$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 0 & 4 & -1 & 5 \\ 1 & 1 & 1 & 6 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

Example of Elimination

$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 0 & 4 & -1 & 5 \\ 1 & 1 & 1 & 6 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

Example of Elimination

$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 0 & 4 & -1 & 5 \\ 1 & 1 & 1 & 6 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & -4 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -11 \end{pmatrix} \quad \left(\begin{array}{ccc|c} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & -4 & -1 & -11 \end{array}\right)$$

Example of Elimination

$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix} \quad \left(\begin{array}{rrr|r} 0 & 4 & -1 & 5 \\ 1 & 1 & 1 & 6 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 1 \end{pmatrix} \quad \left(\begin{array}{rrr|r} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 2 & -2 & 1 & 1 \end{array}\right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & -4 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -11 \end{pmatrix} \quad \left(\begin{array}{rrr|r} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & -4 & -1 & -11 \end{array}\right)$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ -6 \end{pmatrix} \quad \left(\begin{array}{rrr|r} 1 & 1 & 1 & 6 \\ 0 & 4 & -1 & 5 \\ 0 & 0 & -2 & -6 \end{array}\right)$$

Grauss Elimination → Gauss Jordan

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

Grauss Elimination → Gauss Jordan

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

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

Grauss Elimination → Gauss Jordan

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

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

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

Grauss Elimination → Gauss Jordan

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

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

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

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

Questions

Matrix Decomposition

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

$$\mathrm{PA} = \mathrm{LDU}$$

- 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 A is a symmetric positive definite then $\mathrm{U} = \mathrm{L}^T$ and D has strictly positive diagonal elements

Solving the matrix system

- Our objective is to solve

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

- Enable use of forward / backward substitution

Square - Full Rank Matrices

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

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

  where
  - E is a diagonal matrix where elements are the eigenvalues of A
  - S is a matrix where the columns are the eigenvectors of 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 $proj_u(v) = \frac{<v,u>}{<u,u>} u$, i.e., project v onto u
- Process is then
- $u_1 = v_1$
- $u_2 = v_2 - proj_{v_1}(v_2)$
- $u_3 = v_3 - proj_{v_1}(v_3) - proj_{v_2}(v_3)$
- $u_k = v_k - \sum_{j=1}^{k-1} proj_{u_j}(v_k)$
- $e_i = \frac{v_i}{||v_i||}$ 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

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

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

    where $\sigma_1 > \cdots > \sigma_k > 0$ and the 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

$$\underset{Independent}{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}} \qquad \underset{Independent?}{\begin{pmatrix} 1.01 & 1.00 & 1.00 \\ 1.00 & 1.01 & 1.00 \\ 1.00 & 1.00 & 1.01 \end{pmatrix}}$$

   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^TA or AA^T
- We can use

$$A^TA = VDV^T$$

  and

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

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

$$A = UDV^T$$

- Note:
  - rank(A) = rank(D) = k
  - colspace(A) = first k columns of U
  - nullspace(A) = first n-k 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.

Sensivity

- 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