CSC338 Winter 2022

Week 4 - Systems of Linear Equations

Ilir Dema

University of Toronto

Feb 4, 2022

What is vector norm?

Definition

- Let V be a vector space. **Norm** on V is a map $\|\| : V \to \mathbb{R}^{\geq 0}$ satifying the following properties:
  (I) $\forall v \in V, \|v\| \geq 0$ and $\|v\| = 0$ if and only if $v = 0$
  (II) $\forall v \in V, \forall a \in \mathbb{R}, \|av\| = |a|\|v\|$
  (III) $\forall v, w \in V, \|v + w\| \leq \|v\| + \|w\|$

Examples

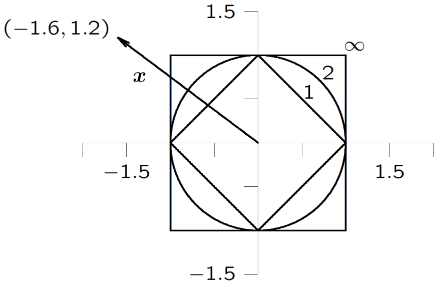
Let $V = \mathbb{R}^n, v = (x_1, \ldots, x_n)$.

- $\|v\|_1 = \sum_{i=1}^n |x_i|$ (1-norm)
- $\|v\|_2 = (\sum_{i=1}^n |x_i|^2)^{1/2}$ (2-norm)
- $\|v\|_\infty = \max_{1 \leq i \leq n} |x_i|$ (∞-norm)

- Exercise: Prove that the above three are indeed norms.
- Exercise: Prove that (in general): $|\|u\| - \|v\|| \leq \|u - v\|$

Geometrical interpreation

Drawing shows unit sphere in two dimensions for each of these norms:



Norms have following values for vector shown:

$$\|\boldsymbol{x}\|_1 = 2.8, \quad \|\boldsymbol{x}\|_2 = 2.0, \quad \|\boldsymbol{x}\|_\infty = 1.6$$

What is matrix norm?

Definition

- Let V be a vector space of dimension n and $A \in M_{n\times n}$ a matrix, representing some linear transformation of V.
- Definition
$$\|A\| = \max_{x\neq 0, x\in V} \frac{\|Ax\|}{\|x\|}$$
- Exercise: Prove that the defined matrix norm satsfies the properties of norms.
- Exercise: Prove that for any two $n \times n$ matrices, $\|AB\| \leq \|A\|\|B\|$.
- Exercise: Prove that for any matrix A and any vector x, $\|Ax\| \leq \|A\|\|x\|$.

Equivalency of norms

Definition

- Two norms $\|\|_1, \|\|_2$ are called equivalent if there exist positive numbers a, b such that $\|x\|_1 \leq a\|x\|_2$ and $\|x\|_2 \leq b\|x\|_1$ for any vector x.

Exercises

- Prove equivalency of $\|\|_1, \|\|_2, \|\|_\infty$ norms in $\mathbb{R}^n$.
- Let A be matrix and $\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|, \|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$. Are they equivalent? Also, are they euqiva,ent to the previously defined matrix norm?
- Exercise: Prove that $\|A\| = \max_{\|x\|=1} \|Ax\|$.

Ill-conditioned systems

Example

- When there is a small change in one or more coefficients in a system, if the system is well conditioned, the change in the solution will also be small
- In the case of ill conditioned systems, a small change in some of the coefficients, will result in large changes in the solution.

$$\begin{bmatrix} 1 & 2 \\ 0.48 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1.47 \end{bmatrix} \implies \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 0.49 & 0.99 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1.47 \end{bmatrix} \implies \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

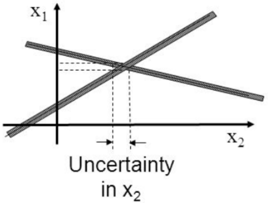
- Q: What is an adequate measure of size for linear systems?

GEOMETRICAL INTERPREATION

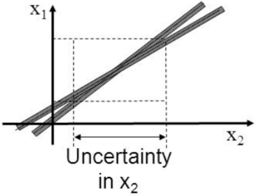
Consider the graphical interpretation for a 2-equation system:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \end{Bmatrix} \qquad \begin{matrix} (1) \\ (2) \end{matrix}$$

We can plot the two linear equations on a graph of x_1 vs. x_2.



Well-conditioned



Ill-conditioned

Conditioning number of a matrix

Definition

- For nonsingular A, $cond(A) = \|A\|\|A^{-1}\|$.
- For singular A, $cond(A) = \infty$.

Properties of $cond(A)$

- For any matrix A, $cond(A) \geq 1$.
- $cond(I) = 1$
- For any nonzero number a, $cond(aA) = cond(A)$
- For a diagonal matrix A, $cond(A) = (\max_i |a_{ii}|)/(\min_i |a_{ii}|)$
- $\|A\|\|A^{-1}\| = (\max_{\|x\|=1} \|Ax\|)(\min_{\|x\|=1} \|Ax\|)^{-1}$, therefore the conditioning number of a matrix describes the ratio to max stretch that A inflicts on unit vectors versus min stretch that A inflicts on unit vectors.

Estimating errors

Definition

- Let x be the true value of a vector, and $\hat{x}$ an approximate value of it. Recall that absolute error of x is the difference $\Delta x = \hat{x} - x$.
- Note that $\|\Delta x\| = \|x - \hat{x}\|$.
- Relative error $\delta x = \frac{\Delta x}{\|x\|}$ will be defined only for nonzero x.

Theorem

Let $Ax = b$ be a linear system. Then the following hold:

$$\|\delta x\| \leq cond(A)\|\delta b\|$$

$$\|\delta x\| \leq cond(A)\|\delta A\|$$

Full backward error

Analysis

Let $Ax = b, \hat{A}\hat{x} = \hat{b}, \hat{A} = A + \Delta A, \hat{x} = x + \Delta x, \hat{b} = b + \Delta b$. It's not hard to see that

$$\Delta Ax + A\Delta x + \Delta A\Delta x = \Delta b$$

hence ingoring second degree differences, we get the linear relationship

$$A\Delta x = \Delta b - \Delta Ax.$$

Since A is invertible,

$$\Delta x = A^{-1}(\Delta b - \Delta Ax).$$

Matrix Conditiong Number

Facts

- Conditioning number of a matrix is not easy to compute because it involves inverting the matrix (a $\mathcal{O}(n^3)$ complexity operation) and computing the norm of both.
- An alternative way (same complexity though!) is to obtain singluar value decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$ where $\boldsymbol{U}, \boldsymbol{V}$ are orthogonal (that is $\det(\boldsymbol{U}), \det(\boldsymbol{V}) = \pm 1$) and $\boldsymbol{\Sigma} = diag(\sigma_i)$. Then $cond(\boldsymbol{A}) = \sigma_1/\sigma_n$.
- An evaluation of $cond(\boldsymbol{A})$ can be obtained by observing that if $\boldsymbol{Ax} = \boldsymbol{y}$ then $\frac{\|\boldsymbol{x}\|}{\|\boldsymbol{y}\|} \leq \|\boldsymbol{A}^{-1}\|$). The equality can be achieved for a careful choice of $\boldsymbol{x}, \boldsymbol{y}$. Usually $\boldsymbol{y}$ is a vector with components ± 1 with signs chosen such that $\|\boldsymbol{x}\|$ is maximized.
- Software packages have functions to compute it, e.g. `numpy.linalg.cond`.

The residual

What is it?

- Suppose a numerical algorithm for $\boldsymbol{Ax} = \boldsymbol{b}$ yields $\hat{\boldsymbol{x}}$.
- One way is to verify our solution, is to plug it in the original system, obtaining $\hat{\boldsymbol{b}} = \boldsymbol{A}\hat{\boldsymbol{x}}$ and evaluate $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}$.
- Some computation (please do it on your own in the form of an exercise!) shows that

$$\frac{\|\Delta x\|}{\|\hat{x}\|} \leq cond(A)\frac{\|r\|}{\|A\|\|\hat{x}\|} \leq cond(A)\frac{\|\Delta A\|}{\|A\|}$$

  where ΔA is the backward error on A.
- This shows residual is good if A is well conditioned.

Residual is no good: example

The residual not good if matrix is not well conditioned

$$\boldsymbol{Ax} = \begin{bmatrix} 0.913 & 0.659 \\ 0.457 & 0.330 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.254 \\ 0.127 \end{bmatrix} = \boldsymbol{b}$$

Consider two approximate solutions

$$\hat{\boldsymbol{x}}_1 = \begin{bmatrix} 0.6391 \\ -0.5 \end{bmatrix}, \hat{\boldsymbol{x}}_2 = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}$$

The norms of their residuals are

$$\|\boldsymbol{r}_1\| = 7.0 \times 10^{-5}, \|\boldsymbol{r}_1\| = 2.4 \times 10^{-2}$$

The real solution is $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. What can we conclude?

Iterative refinement

Here is how

- Recall the residual $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\hat{\boldsymbol{x}}$. The process can be repeated solving $\boldsymbol{r} = \boldsymbol{A}\hat{\boldsymbol{z}}$ and forming the approximate solution $\boldsymbol{x} + \boldsymbol{z}$.
- Denoting $\boldsymbol{x_n}$ the solution obtained in the iteration n, it's not hard to see

$$\|\boldsymbol{x_{n+1}} - \boldsymbol{x}\| \leq \|\boldsymbol{A}^{-1}\| \|\boldsymbol{\Delta A}\| \|\boldsymbol{x_n} - \boldsymbol{x}\|.$$

  where $\boldsymbol{\Delta A}$ is the error in the n-th iteration of Gauss Elimination.
- Clearly, if $\|A^{-1}\| \|\Delta A\| < 1$ the procedure converges.
- It can be shown $\|\boldsymbol{\Delta A}\| \leq \rho \boldsymbol{n} \varepsilon_{\boldsymbol{mach}} \|\boldsymbol{A}\|$ where ρ is a constant factor computed during the Gauss elimination process.

Sherman-Morrison formula

Sherman-Morrison formula gives inverse of matrix resulting from rank-one change to matrix whose inverse is already known:

$$(\boldsymbol{A}-\boldsymbol{u}\boldsymbol{v}^T)^{-1} = \boldsymbol{A}^{-1}+\boldsymbol{A}^{-1}\boldsymbol{u}(1-\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u})^{-1}\boldsymbol{v}^T\boldsymbol{A}^{-1},$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are n-vectors

Evaluation of formula requires $\mathcal{O}(n^2)$ work (for matrix-vector multiplications) rather than $\mathcal{O}(n^3)$ work required for inversion

Sherman-Morrison Example

Consider rank-one modification

$$\begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -1 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$$

(with $3, 2$ entry changed) of system whose LU factorization was computed in earlier example

One way to choose update vectors:

$$\boldsymbol{u} = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{v} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

so matrix of modified system is $\boldsymbol{A} - \boldsymbol{u}\boldsymbol{v}^T$

Example (continued)

Using LU factorization of $\boldsymbol{A}$ to solve $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{u}$ and $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$,

$$\boldsymbol{z} = \begin{bmatrix} -3/2 \\ 1/2 \\ -1/2 \end{bmatrix} \quad \text{and} \quad \boldsymbol{y} = \begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix}$$

Final step computes updated solution

$$\boldsymbol{x} = \boldsymbol{y} + \frac{\boldsymbol{v}^T\boldsymbol{y}}{1 - \boldsymbol{v}^T\boldsymbol{z}}\boldsymbol{z} =$$

$$\begin{bmatrix} -1 \\ 2 \\ 2 \end{bmatrix} + \frac{2}{1 - 1/2}\begin{bmatrix} -3/2 \\ 1/2 \\ -1/2 \end{bmatrix} = \begin{bmatrix} -7 \\ 4 \\ 0 \end{bmatrix}$$

Proof

To solve linear system $(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{v}^T)\boldsymbol{x} = \boldsymbol{b}$ with new matrix, use formula to obtain

$$\begin{aligned} \boldsymbol{x} &= (\boldsymbol{A} - \boldsymbol{u}\boldsymbol{v}^T)^{-1}\boldsymbol{b} \\ &= \boldsymbol{A}^{-1}\boldsymbol{b} + \boldsymbol{A}^{-1}\boldsymbol{u}(1 - \boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{u})^{-1}\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{b}, \end{aligned}$$

which can be implemented by steps

1. Solve $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{u}$ for $\boldsymbol{z}$, so $\boldsymbol{z} = \boldsymbol{A}^{-1}\boldsymbol{u}$
2. Solve $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{b}$ for $\boldsymbol{y}$, so $\boldsymbol{y} = \boldsymbol{A}^{-1}\boldsymbol{b}$
3. Compute $\boldsymbol{x} = \boldsymbol{y} + ((\boldsymbol{v}^T\boldsymbol{y})/(1 - \boldsymbol{v}^T\boldsymbol{z}))\boldsymbol{z}$

Cholesky Factorization

Symmetric Positive Definite Matrices

If $\boldsymbol{A}$ is symmetric and positive definite, then LU factorization can be arranged so that

$$\boldsymbol{U} = \boldsymbol{L}^T, \quad \text{that is,} \quad \boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T,$$

where $\boldsymbol{L}$ is lower triangular with positive diagonal entries

Algorithm for computing *Cholesky factorization* derived by equating corresponding entries of $\boldsymbol{A}$ and $\boldsymbol{L}\boldsymbol{L}^T$ and generating them in correct order

Cholesky Factorization (continued)

Features of Cholesky algorithm symmetric positive definite matrices:

- All n square roots are of positive numbers, so algorithm well defined
- No pivoting required for numerical stability
- Only lower triangle of $\boldsymbol{A}$ accessed, and hence upper triangular portion need not be stored
- Only $n^3/6$ multiplications and similar number of additions required

Symmetric Matrices

For symmetric indefinite A, Cholesky factorization not applicable, and some form of pivoting generally required for numerical stability

Factorization of form

$$PAP^T = LDL^T,$$

with L unit lower triangular and D either tridiagonal or block diagonal with 1×1 and 2×2 diagonal blocks, can be computed stably using symmetric pivoting strategy

References

Michael T. Heath
Scientific Computing (Revised Second Edition)
SIAM