Numerical Matrix Analysis

Notes #2

Linear Algebra Introduction / Review

Peter Blomgren
⟨blomgren@sdsu.edu⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2024
(Revised: February 20, 2024)

Outline

Student Learning Targets, and Objectives

Target Linear Algebra Fundamentals — "Objects"
- **Objective** Vectors in $\mathbb{R}^n$ and $\mathbb{C}^n$
- **Objective** Real $\mathbb{R}^{m\times n}$ and Complex $\mathbb{C}^{m\times n}$ matrices
- **Objective** Vandermonde Matrix; connection to polynomials and linear least-squares (LLSQ)
- **Objective** Range (image) and Nullspace (kernel) of a matrix; matrix rank

Target Linear Algebra Fundamentals — "Actions"
- **Objective** Matrix-Vector Product (two points of view); linearity
- **Objective** Matrix-Matrix Producs
- **Objective** Matrix Transpose

Target Linear Algebra Fundamentals — "Properties"
- **Objective** Equivalent Statements for Invertible Matrices

Linear Algebra: Introduction / Review / Crash Course

We start off by a quick review(?) of basic linear algebra concepts and algorithms.

Depending on your background this will either be a review of things you know, possibly in a new notation / framework, or a crash course.

• • • • •

An n-dimensional **vector** $\vec{x}$ is an n-tuple* of either real $\vec{x} \in \mathbb{R}^n$ or complex $\vec{x} \in \mathbb{C}^n$ numbers, in this class all vectors are **column vectors**, *i.e.*

$$\vec{x} \in \mathbb{R}^n \;\Rightarrow\; \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \text{where } x_i \in \mathbb{R},\; i = 1, 2, \dots, n.$$

* In python "tuples" (1,2,3) are immutable objects, "lists" [1,2,3] are mutable; so generally you want a python list to represent a vector.

Vectors: Transpose, Addition & Subtraction

We express a **row vector** using the **transpose**, *i.e.*

$$\vec{x} \in \mathbb{R}^n \;\Rightarrow\; \vec{x}^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}.$$

Vector addition and subtraction

$$\vec{x}, \vec{y} \in \mathbb{R}^n \;\Rightarrow\; \vec{x} \pm \vec{y} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \pm \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 \pm y_1 \\ x_2 \pm y_2 \\ \vdots \\ x_n \pm y_n \end{bmatrix},$$

or $\vec{z} = \vec{x} + \vec{y}$ where

$$z_i = x_i + y_i, \;\; i = 1, 2, \dots, n.$$

Comment: In [MATH 524] we use $\mathbb{F}$ as a placeholder for "either $\mathbb{R}$ or $\mathbb{C}$", since both are *fields*. We are not adopting this notation for this class, and will occasionally use $\mathbb{F}$ to represent finite-precision *floating-point numbers*.

Matrices — Matrix-Vector Product

An $(m \times n)$ matrix (m rows, n columns) A with real or complex entries is represented

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}, \quad \left\{ \begin{array}{l} a_{ij} \in \mathbb{R}, \text{ or} \\ a_{ij} \in \mathbb{C} \end{array} \right.$$

We write $A \in \mathbb{R}^{m \times n}$ (or $A \in \mathbb{C}^{m \times n}$.)

If $A \in \mathbb{R}^{m \times n}$ and $\vec{x} \in \mathbb{R}^n$, then the **matrix-vector product**, $\vec{b} = A\vec{x}$, is well defined, and $\vec{b} \in \mathbb{R}^m$, where

$$b_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, 2, \dots, m.$$

Matrix-Vector Product... ...Functional Definition

$$\begin{bmatrix} b_1 \\ \mathbf{b_2} \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ \mathbf{a_{21}} & \mathbf{a_{22}} & \mathbf{a_{23}} & \dots & \mathbf{a_{2n}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} \mathbf{x_1} \\ \mathbf{x_2} \\ \mathbf{x_3} \\ \vdots \\ \mathbf{x_n} \end{bmatrix}$$

$$\mathbf{b_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n}$$

Note: n multiplications and $(n-1)$ additions are needed to compute each entry in $\vec{b}$. In total $(m \cdot n)$ multiplications and $(m \cdot (n-1))$ additions are performed. We say that the matrix-vector product requires $\mathcal{O}(\mathbf{m \cdot n})$ operations. Here, we are interpreting the matrix-vector product as a sequence of inner/dot-products of the rows of A and $\vec{x}$.

Note: This is the most "natural" definition for computational purposes...

Matrix-Vector Product... ...as a Linear Combination

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

$$= x_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} + x_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} + x_3 \begin{bmatrix} a_{13} \\ a_{23} \\ \vdots \\ a_{m3} \end{bmatrix} + \cdots + x_n \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

$$= x_1 \vec{a}_1 + x_2 \vec{a}_2 + x_3 \vec{a}_3 + \cdots + x_n \vec{a}_n$$

Note: In most settings, this is the best definition for "intellectual" purposes...

Matrix-Vector Product: Linearity

The map $\vec{x} \mapsto A\vec{x}$ (from $\mathbb{R}^n$ to $\mathbb{R}^m$, or from $\mathbb{C}^n$ to $\mathbb{C}^m$) is **linear**, *i.e.* $\forall \vec{x}, \vec{y} \in \mathbb{R}^n$ $(\mathbb{C}^n)$, and $\alpha, \beta \in \mathbb{R}$ $(\mathbb{C})$

$$\begin{aligned} A(\vec{x}+\vec{y}) &= A\vec{x} + A\vec{y} \\ A(\alpha\vec{x}) &= \alpha A\vec{x} \\ A(\alpha\vec{x}+\beta\vec{y}) &= \alpha A\vec{x} + \beta A\vec{y} \end{aligned}$$

Note: Every linear map from $\mathbb{R}^n$ to $\mathbb{R}^m$ can be expressed as multiplication by an $(m \times n)$-matrix.

More generally, every linear map from a vector space to another vector space, can — given bases for the two spaces — be described by a matrix. [MATH 524]

Example: The Vandermonde Matrix

Given a set of points $\{x_1, x_2, \ldots, x_m\}$, we can express the evaluation of the polynomial

$$p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1}$$

at those points using the $(m \times n)$ **Vandermonde matrix** A, and the vector $\vec{c}$, containing the polynomial coefficients

$$A = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{n-1} \end{bmatrix}, \qquad \vec{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{bmatrix}$$

Forming $\vec{p} = A\vec{c}$ gives us an m-vector containing the values of $p(x_i)$, $i = 1, 2, \ldots, m$.

The Vandermonde Matrix... ...Linear Least Squares

Evaluating polynomials using matrix notation may seem **cute and useless?!?**

But, wait a minute — this notation looks vaguely familiar from the discussion of **linear least squares** (LLSQ) problems from [MATH 541]$^{\text{RIP}}$ (or possibly [MATH 340]).

In case you forgot (or never studied) LLSQ: The goal is to find the best model in a class (*i.e.* low-dimensional polynomials) to measured data (observations y_i, made at the points x_i).

The discrepancy (error) between the model and the observations is measured in the sum-of-squares norm.

Linear Least Squares: Explicit Example 1 of 4

Find the best straight line $p(x) = c_0 + c_1 x$ fitting the observations $(x, y) \in \{(0,1), (1,2), (2,2.5), (3,4), (4,7)\}$.

We have the (5×2) Vandermonde matrix $A = \begin{bmatrix} \vec{1} & \vec{x} \end{bmatrix}$, the 2-vector $\vec{c}$ (of polynomial coefficients) and the 5-vector $\vec{y}$ (of measurements):

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \qquad \vec{c} = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}, \qquad \vec{y} = \begin{bmatrix} 1 \\ 2 \\ 2.5 \\ 4 \\ 7 \end{bmatrix}$$

The **Linear Least Squares Problem:** *Find the $\vec{c}$ which minimizes the least squares error $\|A\vec{c} - \vec{y}\|_2^2$.*

Linear Least Squares: Explicit Example

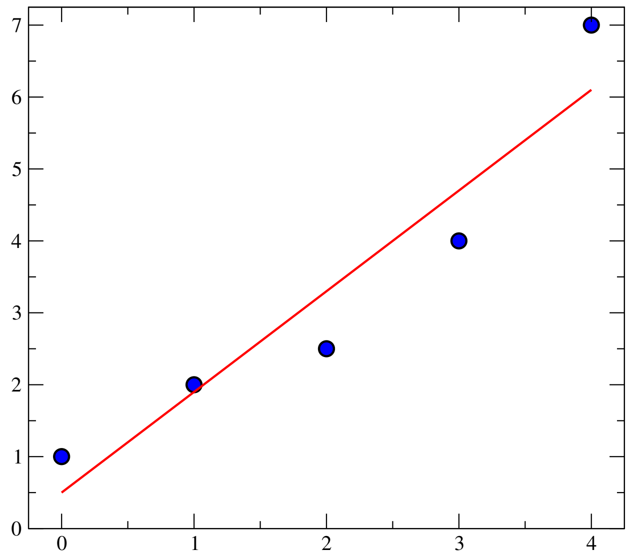

Figure: The data points (x_i, y_i) and the straight line corresponding to the best fit (in the least-squares-sense), *i.e.* $p^*(x) = c_0^* + c_1^* x$.

Linear Least Squares: Explicit Example

Given a model $\vec{c}$, we can evaluate to corresponding linear polynomial $p(x) = c_0 + c_1 x$ at the points x_i: $\vec{p} = A\vec{c}$. The **pointwise error** in the model is $\vec{e} = \vec{p} - \vec{y}$:

$$\vec{p} = \begin{bmatrix} c_0 + 0c_1 \\ c_0 + 1c_1 \\ c_0 + 2c_1 \\ c_0 + 3c_1 \\ c_0 + 4c_1 \end{bmatrix}, \qquad \vec{e} = \begin{bmatrix} c_0 + 0c_1 - 1 \\ c_0 + 1c_1 - 2 \\ c_0 + 2c_1 - 2.5 \\ c_0 + 3c_1 - 4 \\ c_0 + 4c_1 - 7.5 \end{bmatrix}$$

The **least squares error** is given by

$$r_{\text{LSQ}} = \|\vec{e}\|_2^2 = \sum_{i=1}^{5} e_i^2 = \|A\vec{c} - \vec{y}\|_2^2$$

Linear Least Squares: Explicit Example

In order to identify the optimal choice of $\vec{c}$, we compute the partial derivatives with respect to the model parameters, and set those expressions to be zero (in order to identify the optimum)

$$\frac{\partial r_{\text{LSQ}}}{\partial c_0} = \frac{\partial r_{\text{LSQ}}}{\partial c_1} = 0.$$

After some work (which is not central to this discussion), we get the **Normal Equations**

$$A^T A\vec{c} = A^T\vec{y} \quad \Leftrightarrow \quad A^T(A\vec{c} - \vec{y}) = 0$$

Even though the matrix A is (usually) tall and skinny (here (5×2)), the matrix $A^T A$ is square; here (2×2). The (formal) solution $\vec{c} = [A^T A]^{-1} A^T \vec{y}$, to this linear system gives us the coefficients for the optimal polynomial (the red line on slide 13).

Unanswered Questions...

The previous LLSQ example raises more questions than it answers, the most important one **"Would anyone in his/her right mind form the matrix $A^T A$, then invert it $[A^T A]^{-1}$, then multiply the vector $A^T\vec{y}$ by the inverse?"**

The answer is **"No!"** ...which raises even more questions!

This class is all about how to solve linear systems... taking issues like *(i)* speed; *(ii)* accuracy; and *(iii)* stability into consideration.

We will revisit the questions raised by the example in more detail later... However, we will use the example to introduce some further linear algebra functionality and terminology...

Matrix-Matrix Product

The matrix-matrix product $B = AC$ is well defined if the matrix C has as many rows as the matrix A has columns

$$B_{k\times n} = A_{k\times m} \cdot C_{m\times n}$$

The elements of B are defined by

$$b_{i\ell} = \sum_{k=1}^{m} a_{ik} c_{k\ell}$$

Sometimes it is useful to think of the columns of B, $\vec{b}_\ell$ as linear combinations of the columns of A:

$$\vec{b}_\ell = A\vec{c}_\ell = \sum_{k=1}^{m} c_{k\ell}\vec{a}_k$$

The Transpose of a Matrix (A^T)

The transpose of a matrix $A = \{a_{ij}\}$ is the matrix $A^T = \{a_{ji}\}$, *e.g.*:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}, \quad A^T = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{12} & a_{22} & a_{32} & a_{42} \\ a_{13} & a_{23} & a_{33} & a_{43} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix}$$

— just mirror across the diagonal — but can be quite (memory-access) expensive, especially for large matrices.

For complex matrices $C = \{c_{ij}\}$, the complex (Hermitian) transpose is given by $C^* = \{c^*_{ji}\}$, where c^* is the complex conjugate of c:

$$c = a + bi, \quad c^* = a - bi.$$

Note: Mathematically, the transpose is a no-op; but if implemented carelessly, it can trigger a lot of data-shuffling.

The Range and Nullspace of a Matrix A

The **range** (or image) of a matrix, written $\mathrm{range}(A)$, is the set of vectors that can be expressed as a linear combination of the columns of $A_{m\times n}$, *i.e.*

$$\mathrm{range}(A) = \{\vec{y} \in \mathbb{R}^m \,:\, \vec{y} = A\vec{x}, \text{ for some } \vec{x} \in \mathbb{R}^n\}$$

we say **"$\mathrm{range}(A)$ is the space spanned by the columns of A."**

The **nullspace** (or kernel) of a matrix A, written $\mathrm{null}(A)$, is the set of vectors that satisfy $A\vec{x} = 0$, *i.e.*

$$\mathrm{null}(A) = \{\vec{x} \in \mathbb{R}^n \,:\, A\vec{x} = 0\}$$

Note: In [MATH 254], we tend to talk about the image and kernel; and in [MATH 524] we lean in the direction of the range – nullspace terminology.

The Rank of a Matrix $A_{m\times n}$

The **column rank** of a matrix is the dimension of $\mathrm{range}(A)$, its "column space." The **row rank** of a matrix is the dimension of its "row space," or $\mathrm{range}(A^T)$.

The column rank is always equal to the row rank (we will see the proof of this in a few lectures), hence we only refer to the **Rank** of a matrix

$$\mathrm{rank}(A)$$

An $(m \times n)$ matrix, $A \in \mathbb{R}^{m\times n}$, is of **full rank** if it has the maximal possible rank $\min(m, n)$.

If an $(m \times n)$ matrix, $A \in \mathbb{R}^{m\times n}$ where $m \geq n$, has full rank; then it must have n **linearly independent columns**.

Recall: The Normal Equations

$$A^T A\vec{c} = A^T\vec{y} \quad \Leftrightarrow \quad A^T(A\vec{c}-\vec{y}) = 0$$

Due to the "tall-and-skinniness" of A, the equation $A\vec{c}-\vec{y}=0$ does not necessarily have a solution.

Given a vector $\vec{c}$ we can define the **residual**, $\vec{r}(\vec{c}) = A\vec{c}-\vec{y}$, which measures how far (point-wise) from solving the system we are.

We notice that the solution to the normal equations requires that the residual is in the **nullspace** of A^T.

The solution is in $\mathrm{range}(A)$ such that the residual is orthogonal (perpendicular) to $\mathrm{range}\left(A^T\right)$.

Note: The solution can also be thought of as the orthogonal projection of $\vec{y}$ onto $\mathrm{range}(A)$. We will adopt this view soon...

The Inverse of a Matrix A

An **invertible** or **nonsingular** matrix A is a square matrix of full rank.

The m columns of an invertible matrix form a **basis** for the whole space $\mathbb{R}^m$ (or $\mathbb{C}^m$) — any vector $\vec{x} \in \mathbb{R}^m$ can be expressed as a **unique** linear combination of the columns of A.

In particular we can express the unit vector $\vec{e}_j$ (which has a 1 in position j and zeros in all other positions):

$$\vec{e}_j = \sum_{i=1}^{m} z_{ij}\vec{a}_i, \quad \Leftrightarrow \quad \vec{e}_j = A\vec{z}_j$$

If we play this game for $j = 1 \ldots m$, we get

$$\underbrace{[\vec{e}_1\ \vec{e}_2\ \ldots\ \vec{e}_m]}_{I_{m\times m}} = A\underbrace{[\vec{z}_1\ \vec{z}_2\ \ldots\ \vec{z}_m]}_{Z}$$

The Inverse of a Matrix A

We have

$$I_{m\times m} = A\cdot Z$$

The $(m\times m)$ matrix $I_{m\times m}$ which has ones on the diagonal and zeros everywhere else is the **identity matrix**.

The matrix Z is the **inverse** of A.

Any square nonsingular matrix A has a unique inverse, written A^{-1}, which satisfies

$$A\cdot A^{-1} = A^{-1}\cdot A = I$$

Equivalent Statements for a Square Matrix $A \in \mathbb{C}^{m\times m}$

For a matrix $A \in \mathbb{C}^{m\times m}$ the following are **equivalent**

- A has an inverse A^{-1}
- The linear system $A\vec{x} = \vec{b}$ has a unique solution $\vec{x}$, $\forall\vec{b} \in \mathbb{R}^m$
- $\mathrm{rank}(A) = m$
- $\mathrm{range}(A) = \mathbb{C}^m$
- $\mathrm{null}(A) = \{\vec{0}\}$
- 0 is not an **eigenvalue** of A
- 0 is not a **singular value** of A
- $\det(A) \neq 0$

Note: The determinant is rarely useful in numerical algorithms — it is usually too expensive to compute.

Homework #1 — Due Date in Canvas/Gradescope

TB-1.2: Suppose the masses m_1, m_2, m_3, m_4 are located at positions x_1, x_2, x_3, x_4 in a line and connected by springs with spring constants k_{12}, k_{23}, k_{34} whose natural lengths of extension are ℓ_{12}, ℓ_{23}, ℓ_{34}. Let f_1, f_2, f_3, f_4 denote the rightward forces on the masses, *e.g.* $f_1 = k_{12}((x_2 - x_1) - \ell_{12})$.

(a) Write the (4×4) matrix equation relating the column vectors $\vec{f}$ and $\vec{x}$. Let K denote the matrix in this equation.

(b) What are the units of the entries of K in the physics sense (*e.g.* mass $\times$ time, distance / mass, etc...)

(c) What are the units of $\det(K)$, again in the physics sense?

(d) Suppose K is given numerical values based on the units meters, kilograms and seconds. Now the system is rewritten with a matrix K' based on centimeters, grams, and seconds. What is the relationship of K' to K? What is the relationship of $\det(K')$ to $\det(K)$?

Illustration: Homework #1

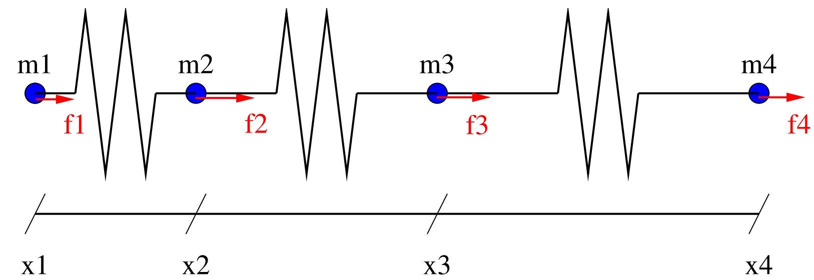

Figure: Note that the illustration is not necessarily complete. In particular, recall Newton's 3rd Law of Motion: *When one body exerts a force on a second body, the second body simultaneously exerts a force equal in magnitude and opposite in direction to that of the first body.*

Notes: You may want to look up **Newton's 3rd Law**, and **Hooke's Law.**

The purpose of this assignment is to (1)remind outselves that matrices and vectors usually describe something "real"; and (2)work on problem-solving skills for a potentially unfamiliar problem.

Homework AI-Policy Spring 2024

AI-era Policies — SPRING 2024

AI-3 **Documented:** *Students can use AI in any manner for this assessment or deliverable, but they must provide appropriate documentation for all AI use.*

This applies to ALL MATH-543 WORK during the SPRING 2024 semester.

The goal is to leverage existing tools and resources to generate HIGH QUALITY SOLUTIONS to all assessments.

You MUST document what tools you use and HOW they were used (including prompts); AND how results were VALIDATED.

BE PREPARED to DISCUSS homework solutions and AI-strategies. **Participation in the in-class discussions will be an essential component of the grade for each assessment.**