INTRODUCTION

MARIA CAMERON

Contents

1. What is data science?

According to Wiki, *data science* is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, machine learning and big data. A very interesting philosophical paper "50 Years of Data Science" by David Donoho offers an extensive discussion on what is Data Science, when was it born, what is its present, and what is its future. The gist of it is that *Data Science is a science about learning from data*. It is a massive explosion of problems and methods that happened as a result of synthesis of statistics and computer science (this is my understanding of it). The activities of Data Science are classified into the following 6 divisions (D. Donoho):

(1) Data Exploration and Preparation;
(2) Data Representation and Transformation;
(3) Computing with Data;

(4) Data Modeling;
(5) Data Visualization and Presentation;
(6) Science about Data Science.

Year 1962 can be considered as the years of birth of Data Science: John Tukey published a paper The Future of Data Analysis in the journal *The Annals of Mathematical Statistics* (Vol. 33, No. 1 (Mar., 1962), pp. 1–67), specializing on definitions, theorems, and rigorous proofs, where he expressed his point that the demand for data gathering, data analysis and interpretation goes far beyond what is offered by the classical statistical "diet". This paper shocked its readers, academic statisticians, and has become highly influential.

2. Data Science, Machine Learning, and Scientific computing

In Spring 2019, David Bindel spent part of his sabbatical in the UMD and delivered a series of 6 lectures entitled "Numerical Methods for Data Science" which was a compression of the course that he developed in Cornell University to make Numerical Methods classes appealing for CS students. This was a really inspirational moment for me. While listening to Bindel, I was thinking: (*i*) why don't we offer such a course? and (*ii*) I can do it and I am really thrilled about doing it despite it will require a lot of learning. I taught this course in Fall 2020 for the first time. Course materials from Fall 2020 and Fall 2021 are available on my personal website: https://www.math.umd.edu/ mariakc/NumericalMethodsforDataScienceAndMachineLearning.html.

I believe we are currently living through an exciting time of a *revolution in computational mathematics* that started in ~2018 and is marked by the rapid development of methods based on techniques borrowed from data science, machine learning, and quantum physics [5]. While the traditional numerical methods for solving partial differential equations (PDEs) based on finite difference and finite element discretization still remain very important in such applications as fluid dynamics and as reliable and well-understood testing techniques, their applicability is limited to low-dimensional problems as their cost grows exponentially with the dimension of the ambient space. Novel numerical methods based on (*i*) exploiting the intrinsic low-dimensional geometry of the problem hidden in the high-dimensional space by means of diffusion maps [6], or (*ii*) the representation power of neural networks [7] and (*iii*) tensor trains [8] allow us to tackle higher-dimensional problems inaccessible to the traditional techniques. Due to the novelty of these methods in computational mathematics, the most basic questions are yet to be answered. How can we control numerical errors? How one can make reasonable choices for the settings of the method under consideration in a systematic way? How does the computational cost scale with the problem size?

The use of artificial neural networks enables the numerical solution to high-dimensional PDEs that used to be infeasible due to the curse of dimensionality. Within the last couple of years, I found that many of my mathematical and scientific friends have harnessed techniques of machine learning and made huge progress in such fields as PDE solving [1], rare events quantification in stochastic processes [2, 3], geophysics, and computational chemistry [4], also see web page of G. Rotskoff.

Novel methods based on reinforcement learning, (see the web page of Prof. Haizhao Yang) aim at approximating solutions to PDEs by formulas of finite complexity and exhibit dimension-independent computational cost.

Meshless kernel-based methods such as diffusion maps-based PDE solvers also are suitable for high dimensions provided that data points occupy a low-dimensional unknown manifold (e.g. papers by Evans, Cameron, and Tiwary).

Quantum algorithms allow for polynomial-time solutions for NP-hard problems such as factorization to prime numbers. Currently, they are a subject of active research. They, however, are beyond the scope of this course.

3. Course syllabus and necessary background

The course will consist of four chapters.

(1) *Optimization for large-scale machine learning.* Some important references: the review paper "Optimization methods for machine learning" by L. Bottou, F. Curtis, and J. Nocedal and the large textbook J. Nocedal and S. Wright "Numerical Optimization";
(2) *Matrix data and latent factor models.* Some important references: D. Bindel's course "Numerical Methods for Data Science" delivered in the summer school in Shanghai (because it has a really helpful set of lecture notes ☺);
(3) *Nonlinear dimensionality reduction.* Some important references: A. Benson's course "Numerical Methods for Data Science" and the paper "Diffusion Maps" by R. Coifman and S. Lafon;
(4) *Numerical methods for graph data analysis.* Some important references: the review paper M. Newman "Structure and function of complex networks" and references therein, A. L. Barabasi Network Science

Some background for this class from linear algebra, calculus, CS, Matlab, floating point arithmetic, and conditioning is compiled in D. Bindel's note.

4. A brief review of vector and matrix norms, basic matrix decompositions, and condition numbers

This review is largely based on D. Bindel's and J. Goodman's online textbook "Principles of Scientific Computing" and J. Demmel's textbook "Applied Numerical Linear Algebra" [9]. A partially debugged version of Bindel&Goodman is available here.

4.1. **Vector spaces.**

Definition 1. *A vector space V is a set closed with respect to the operations of addition "$+$": $V \times V \to V$, and multiplication by a scalar "$\cdot\alpha$": $V \to V$. The operations satisfy the*

following properties.

(1) $\mathbf{a}+\mathbf{b}=\mathbf{b}+\mathbf{a}$,
(2) $(\mathbf{a}+\mathbf{b})+\mathbf{c}=\mathbf{a}+(\mathbf{b}+\mathbf{c})$,
(3) $\alpha(\mathbf{a}+\mathbf{b})=\alpha\mathbf{a}+\alpha\mathbf{b}$,
(4) $(\alpha+\beta)\mathbf{a}=\alpha\mathbf{a}+\beta\mathbf{a}$,
(5) there is $0\in V$ s.t. $\mathbf{a}+\mathbf{0}=\mathbf{a}$ for any $\mathbf{a}\in V$,
(6) for any $\mathbf{a}\in V$ there is $(-\mathbf{a})\in V$ s.t. $\mathbf{a}+(-\mathbf{a})=0$,
(7) $\alpha(\beta\mathbf{a})=(\alpha\beta)\mathbf{a}$,
(8) $1\mathbf{a}=\mathbf{a}$ for any $\mathbf{a}\in V$.

Exercise Prove that for any $\mathbf{a}\in V$ $0\mathbf{a}=\mathbf{0}$ where $0\in\mathbb{R}$ while $\mathbf{0}\in V$.

Below we remind some basic concepts. Please read Sections 4.2.1 and 4.2.2 in Bindel&Goodman for more details.

- A *subspace* W of a vector space V is a subset of V that is a vector space itself with respect to the same operations as in V, i.e., W is closed under addition and scalar multiplication: for any $w_1, w_2\in W$ and $\alpha\in\mathbb{R}$ or $\mathbb{C}$, $w_1+W_2\in W$ and $\alpha w_1\in W$. Therefore, to check if W is a subspace, it suffices to check if it closed under addition and scalar multiplication. The properties of the operations are inherited for those in V.
- The span of vectors v_1, ..., v_n in V is the set of their all possible linear combinations.
- We say that vectors v_1, ..., v_n are *linearly independent* if any their zero linear combination implies that all of its coefficients are zero.
- A *basis* of V is a subset of vectors $\{b_i\}_{i\in\mathcal{I}}$ such that:
  (1) any $v\in V$ can be represented as
  $$v=\sum_{i\in\mathcal{I}}\alpha_i b_i,$$
  (2) and the $\{b_i\}_{i\in\mathcal{I}}$ is minimal in the sense such that for any $m\in\mathcal{I}$ one can find $v\in V$ such that
  $$v-\sum_{i\in\mathcal{I}\setminus m}\alpha_i b_i\neq 0$$
  for any set of values of α_i, $i\in\mathcal{I}\setminus\{m\}$.
  Recall a theorem in linear algebra saying that if there is a basis in V $\{b_i\}_{i=1}^n$, then any other basis in V also has n vectors.
- If the number of vectors in a basis of V is finite, this number is called the *dimension* of V. Otherwise, the vector space is *infinitely dimensional*.
- A *linear transformation* or a *linear map* for a vector space V to a vector space W is a map $L: V\to W$ such that for any $v_1, v_2\in V$ and any $\alpha\in\mathbb{R}$ or $\mathbb{C}$
  $$L(v_1+v_2)=L(v_1)+L(v_2)\quad\text{and}\quad L(\alpha v_1)=\alpha L(v_1).$$

Let $\mathcal{B} = \{b_i\}$ be a basis in V and $\mathcal{E} = \{e_i\}$ be a basis in W. Then by linearity we have:

$$L(v) = L\left(\sum_j v_j b_j\right) = \sum_j v_j L(b_j) = \sum_j v_j \sum_i a_{ij} e_i \quad \text{where} \quad L(b_j) = \sum_i a_{ij} e_i.$$

Therefore, we can define the matrix of the linear transformation

$$A = {}_{\mathcal{E}}[L]_{\mathcal{B}} = (a_{ij}).$$

Its columns are the images of the basis vectors in V written in the basis in W.

- A matrix product AB is defined if and only if the number of columns in A is equal the number of rows in B. The matrix product AB corresponds to a composition of linear transformations with matrices A and B. Matrix multiplication is associative but not commutative.
- For a matrix $A = (a_{ij})$ the *transpose* is defined by $A^\top := (a_{ji})$. If A has complex entries, than its *adjoint* is defined as its transpose with complex conjugation: $A^* := (\bar{a}_{ji})$.

Now let us list some examples illustrating these concepts.

Example (1) $\mathbb{R}^n$ is an n-dimensional vector space. Its standard basis is $\{e_i\}$ where e_i is a vector with entry 1 at the ith place and the rest of entries being zeros. Its subset of vectors satisfying $\sum_{i=1}^n a_i = 0$ is an $n-1$-dimensional subspace, while the subset of vectors satisfying $\sum_{i=1}^n a_i = 1$ is not a subspace as it is not closed under addition and scalar multiplication.

(2) The set of polynomials of degree less or equal than n denoted by $\mathcal{P}_n$ is an $(n+1)$-dimensional vector space. One basis in it is the set

$$\mathcal{X} := \{1, x, \ldots, x^n\}.$$

(3) An example of linear transformation from $\mathcal{P}_n$ to $\mathcal{P}_{n-1}$ is the differentiation:

$$\frac{d}{dx} : \mathcal{P}_n \to \mathcal{P}_{n-1}.$$

Its matrix in the basis $\mathcal{X}$ is

$$D_{\mathcal{X}} := \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 2 & \ldots & \\ & & & \ddots & \\ 0 & \ldots & & 0 & n \end{bmatrix}.$$

If we pick another basis, for example, Chebyshev's basis, the differentiation matrix will be different.

(4) Example of an infinite-dimensional space is the space of all polynomials, the space of all continuous functions on an interval $[a, b]$, the space of all continuous functions on $[a, b]$ satisfying the homogeneous boundary conditions $f(a) = f(b) = 0$, etc.

4.2. **Vector norms.**

Definition 2. *Norm is a function defined on a vector space V:*

$$\mathcal{N} : V \longrightarrow \overline{\mathbb{R}}_+ \equiv [0, +\infty]$$

such that

$$(1)\ \|\mathbf{a}\| \geq 0, \quad \|\mathbf{a}\| = 0 \ \text{ iff } \ \mathbf{a} = \mathbf{0},$$
$$(2)\ \|\alpha\mathbf{a}\| = |\alpha|\|\mathbf{a}\|,$$
$$(3)\ \|\mathbf{a} + \mathbf{b}\| \leq \|\mathbf{a}\| + \|\mathbf{b}\|.$$

Example The space of continuous functions on the interval $[a, b]$ with the maximum norm

$$V = C([a, b]), \quad \|f\| = \sup_{[a,b]} |f(x)|.$$

If the interval is finite, $\|f\| = \max_{[a,b]} |f(x)|$.

Example The space of continuous functions on the interval $[a, b]$ with the maximum norm

$$V = L_p([a, b]), \quad \|f\| = \left(\int_a^b |f(x)|^p dx \right)^{1/p}.$$

Example The space $V = l_p$ of all sequences $\{a_k\}_{k=1}^{\infty}$ such that

$$\|\{a\}\| := \left(\sum_{k=1}^{\infty} |a_k|^p \right)^{1/p} < \infty.$$

In particular, l_1 is the space of all absolutely convergent sequences as

$$|a| := \sum_{k=1}^{\infty} |a_k| < \infty.$$

Example The space $V = l_\infty$ of all sequences $\{a_k\}_{k=1}^{\infty}$ such that

$$\|\{a\}\| := \sup_k |a_k| < \infty.$$

In other words, l_∞ is the space of all bounded sequences.

The concept of orthogonality is generalized to vector spaces via the notion of the inner product.

Definition 3. *An inner product is a function $(\cdot, \cdot) : V \times V \longrightarrow \mathbb{R}$ or $\mathbb{C}$ satisfying*

$$(1)\ (\mathbf{a}, \mathbf{a}) \geq 0, \quad (\mathbf{a}, \mathbf{a}) = 0 \text{ iff } \mathbf{a} = \mathbf{0},$$
$$(2)\ (\mathbf{a}, \mathbf{b}) = \overline{(\mathbf{b}, \mathbf{a})},$$
$$(3)\ (\mathbf{a}, \mathbf{b} + \mathbf{c}) = (\mathbf{a}, \mathbf{b}) + (\mathbf{a}, \mathbf{c}),$$
$$(4)\ (\alpha\mathbf{a}, \mathbf{b}) = \alpha(\mathbf{a}, \mathbf{b}).$$

The norm induced by an inner product is given by $\|f\| = \sqrt{(f, f)}$. The norms that are associated with inner products are especially important.

Example (1)

$$f, g \in L_2([a, b]), \quad (f, g) = \int_a^b f(x)\overline{g(x)} dx.$$

(2) Chebyshev inner product.

$$f, g \in C([-1, 1]), \quad (f, g) = \int_{-1}^{1} \frac{f(x)g(x)}{\sqrt{1 - x^2}} dx.$$

Suppose we are looking at the error $e(x) = f(x) - p(x)$ where f is a given function and p is its approximation. The Chebyshev norm puts more weight to the ends of the interval, i.e., the error near the ends of the interval contributes more to the norm than the error near its midpoint.

(3) Hermite inner product.

$$f, g \in C([-\infty, \infty]), \quad (f, g) = \int_{-\infty}^{\infty} f(x) g(x) e^{-x^2} dx.$$

Suppose we are looking at the error $e(x) = f(x) - p(x)$ where f is a given function and p is its approximation. Only the error around the origin will contribute significantly to the norm.

4.3. **Matrix norm.**

Definition 4. *The norm of a matrix associated with the vector norm $\|\cdot\|$ is defined as*

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}. \tag{1}$$

The geometric sense of the matrix norm is the maximal elongation of a unit vector as a result of the corresponding linear transformation.

Exercise Let $A = (a_{ij})$ be an $m \times n$ matrix, $m \geq n$. Show that then:

(1) For the l_1-norm,

$$\|A\|_1 = \max_j \sum_i |a_{ij}|,$$

i.e., the maximal column sum of absolute values.

(2) For the max-norm or l_∞-norm

$$\|A\|_{\max} = \max_i \sum_j |a_{ij}|,$$

i.e., the maximal row sum of absolute values

4.4. **Eigenvalues and eigenvectors.** Finding eigenvalues and eigenvectors is often very useful in many different contexts. For example, the general analytic solution to a linear system of ODEs $\dot{x} = Ax$ is often written in terms of eigenvalues and eigenvectors of A. The 2-norm of A is expressed in terms of eigenvalues of $A^\top A$.

4.4.1. *Diagonalizable matrices.* Recall that an $n \times n$ matrix A is called *diagonalizable* if it has n linearly independent eigenvectors. In this case, A can we written as

$$A = R\Lambda R^{-1} \equiv R\Lambda L = \begin{bmatrix} r_1 & r_2 & \cdots & r_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \begin{bmatrix} l_1 & \rightarrow \\ l_2 & \rightarrow \\ \vdots & \\ l_n & \rightarrow \end{bmatrix}. \tag{2}$$

The columns of R are the right eigenvectors of A. They satisfy:

$$Ar_j = \lambda_j r_j.$$

The rows of $L := R^{-1}$ are the left eigenvectors of A satisfying

$$l_j A = \lambda_j l_j.$$

Even if A is real, eigenvectors and eigenvalues do not need to be real: they are complex in the general case.

4.4.2. *Symmetric matrices.* In the special case where A is real and symmetric, there always exists an orthonormal basis of real eigenvectors, the eigenvalues are real, and the eigenvectors corresponding to distinct eigenvalues are orthogonal.

Exercise Let $A = (a_{ij})$ be an $m \times n$ matrix, $m \geq n$. Show that then for the vector l_2-norm,

$$\|A\|_2 = \sqrt{\rho(A^\top A)}.$$

Solution Recall that the vector 2-norm is given by $\|x\|_2 = \sqrt{x^\top x}$. Using this we get

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{x^\top x=1} \sqrt{x^\top A^\top A x} = \max_{x^\top x=1} \sqrt{x^\top A^2 x}.$$

Since $A^\top A$ is symmetric, its eigen-decomposition is given by

$$A^\top A = U \Lambda U^\top,$$

where U is an orthogonal matrix (i.e., $U^\top U = UU^\top = I$, or $U^\top = U^{-1}$) whose columns are the eigenvectors of A, and Λ is a diagonal matrix whose diagonal entries are the corresponding eigenvalues. Using this we continue:

$$\|A\|_2 = \max_{x^\top x=1} \sqrt{x^\top U \Lambda U^\top x} = \max_{x^\top x=1} \sqrt{(U^\top x)^\top \Lambda (U^\top x)}.$$

Now we note that

$$\|x\|_2 = \|U^\top x\|_2$$

because

$$\|x\|_2^2 = x^\top x = x^\top U U^\top x = (U^\top x)^\top (U^\top x) = \|U^\top x\|^2.$$

Let us denote $U^\top x$ by y. Then

$$\|A\|_2 = \max_{y^\top y=1} \sqrt{y^\top \Lambda y} = \max_{y^\top y=1} \sqrt{y_1^2 \lambda_1 + y_2^2 \lambda_2 + \ldots + y_n^2 \lambda_n} = \max_{j=1,\ldots,n} \sqrt{|\lambda_n|} \equiv \sqrt{\rho(A^\top A)}.$$

Remark If A is a real symmetric matrix, then the eigenvalues of $A^\top A$ are squares of the eigenvalues of A. Hence the 2-norm of A is the spectral radius of A:

$$\|A\|_2 = \max_i |\lambda_i| = \rho(A),$$

4.4.3. *Defective matrices and the Jordan form.* If matrix is not diagonalizable, it is called *defective*. An example of such a matrix is

$$A = \begin{bmatrix} 1 & 10 \\ 0 & 1 \end{bmatrix}. \tag{3}$$

This matrix has eigenvalue 1 of *algebraic multiplicity* two and just one eigenvector $[1, 0]^\top$. This means that the *geometric multiplicity* of eigenvalue 1 is one. In linear algebra, the Jordan form is often considered for such matrices:

$$A = V J V^{-1} \tag{4}$$

where J is a block-diagonal matrix with blocks of the form

$$J_j := \begin{bmatrix} \lambda_j & 1 & & & \\ & \lambda_j & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_j & 1 \\ & & & & \lambda_j \end{bmatrix}.$$

There is a unique eigenvector v_j corresponding to each block. The columns of V form the Jordan basis.

In numerical linear algebra, the Jordan form is rarely computed. The reason is that it is unstable with respect to small perturbations of A. For example, consider a 16×16 matrix A

$$A := \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}. \tag{5}$$

It is already in the Jordan form consisting of a single block, and its unique eigenvalue of algebraic multiplicity 16 is zero. Indeed,

$$\det(\lambda I - A) = \lambda^{16} = 0.$$

Now consider a perturbation of A such that the zero at its bottom left corner is replaced with 10^{-16}:

$$A + \delta A := \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ 10^{-16} & & & & 0 \end{bmatrix}. \tag{6}$$

The eigenvalues of $A + \delta A$ are the roots of

$$\det(\lambda I - A) = \lambda^{16} - 10^{-16} = 0.$$

There are 16 distinct complex eigenvalues located at the corners of the 16-gon in the complex plane:

$$\lambda_k = 0.1 e^{i 2\pi k/16}, \quad k = 0, 1, \ldots 15.$$

Hence, the Jordan form of A will be $\mathrm{diag}\{\lambda_0, \ldots, \lambda_{15}\}$ which is not close to (6). Thus, we see that a perturbation of the size of the machine epsilon has a dramatic effect on the Jordan form and on the magnitudes of the eigenvalues of A.

4.4.4. *The Schur form.* For reasons indicated in Section 4.4.3 the Jordan form of a matrix is rarely computed. Another eigenvalue revealing form is much more preferable: the Schur form defined by:

$$A = QTQ^{\top}$$

where T is upper-triangular,

$$T = \begin{bmatrix} \lambda_1 & t_{12} & t_{13} & \ldots & t_{1n} \\ & \lambda_2 & t_{23} & \ldots & t_{2n} \\ & & \ddots & \ddots & \\ & & & \lambda_{n-1} & t_{n-1,n} \\ & & & & \lambda_n \end{bmatrix}.$$

and Q is orthogonal (or unitary if it is complex), i.e., its columns form an orthonormal basis, or $Q^*Q = I$. Often it is more preferable to deal with the so-called real Schur form in which complex pairs of eigenvalues form 2×2 blocks along the diagonal of T. Then both Q and T are real. The Matlab command to compute the Schur form is

```
A = rand(10);
[Q,T] = schur(A);
```

If A is real, this command computes the real Schur form. If you would like the complex Schur form, type

```
[Q,T] = schur(A,'complex');
```

Exercise Let $u + iv$ be a complex eigenvector of a real matrix A, and $\mu + i\nu$ be the corresponding eigenvalue. Show that

$$A[u, v] = [u, v]\begin{bmatrix} \mu & \nu \\ -\nu & \mu \end{bmatrix}, \tag{7}$$

i.e., the vectors u and v span a 2-dimensional invariant subspace of A.

4.5. **QR and SVD.** Here we briefly remind the QR and SVD matrix decompositions (see [9]). We will return to them in more details later.

Theorem 1. *Let A be $m \times n$, $m \geq n$. Suppose that A has full column rank. Then there exist a unique $m \times n$ orthogonal matrix Q, i.e., $Q^\top Q = I_{n\times n}$, and a unique $n \times n$ upper-triangular matrix R with positive diagonals $r_{ii} > 0$ such that $A = QR$.*

Proof. The proof of this theorem is given by the Gram-Schmidt orthogonalization process.

Algorithm 1: Gram-Schmidt orthogonalization

Input : matrix $A = [a_1\ a_2\ \dots\ a_n]$, $m \times n$, $\operatorname{rank}(A) = n$.
Output: orthogonal matrix Q $m \times n$, $Q^\top Q = I_{n\times n}$, and upper-triangular $n \times n$ matrix R with $r_{ii} > 0$.

for $i = 1, \dots, N$ **do**
  $q_i = a_i$;
  for $j = 1, \dots, i-1$ **do**
    $\begin{cases} r_{ji} = q_j^\top a_i & \text{CGS} \\ r_{ji} = q_j^\top q_i & \text{MGS} \end{cases}$;
    $q_i = q_i - r_{ji} q_j$;
  end
  $r_{ii} = \|q_i\|$;
  $q_i = q_i / r_{ii}$;
end

Here CGS and MGS stand for the Classic Gram-Schmidt and the Modified Gram-Schmidt respectively. □

Unfortunately the classic Gram-Schmidt algorithm is numerically unstable when the columns of A are nearly linearly dependent. The modified Gram-Schmidt is better but still can result in Q that is far from orthogonal (i.e., $\|Q^\top Q - I\|$ is much larger than the machine ϵ) when A is ill-conditioned. There are numerically stable ways to compute the QR-decomposition, i.e., by using the Householder reflections or Givens' rotations.

The Singular Value Decomposition (SVD) shows that any linear transformation from one vector space to another can be represented as a diagonal matrix as soon as right bases are chosen in each of these spaces.

Theorem 2. [9] *Let A be an arbitrary $m \times n$ matrix with $m \geq n$. Then we can write*

$$A = U\Sigma V^\top,$$

where

$$U \text{ is } m \times n \quad \text{and} \quad U^\top U = I_{n\times n},$$
$$\Sigma = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\}, \quad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0,$$
$$\text{and} \quad V \text{ is } n \times n \quad \text{and} \quad V^\top V = I_{n\times n}.$$

The columns of U, u_1, ..., u_n, are called left singular vectors. The columns of V, v_1, ..., v_n are called right singular vectors. The numbers σ_1, ..., σ_n are called singular values. If $m < n$, the SVD is defined for $A^\top$.

The geometric sense of this theorem is the following. Let us view the matrix A as a map from $\mathbb{R}^n$ into $\mathbb{R}^m$:

$$A : \mathbb{R}^n \to \mathbb{R}^m, \quad x \mapsto Ax.$$

Then one can find orthogonal bases in $\mathbb{R}^n$, v_1, ... ,v_n, and in $\mathbb{R}^m$, u_1, ..., u_m and numbers σ_1, ..., σ_n, such that

$$v_j \mapsto \sigma_j u_j, \quad j = 1, \ldots, n.$$

Then for any $x \in \mathbb{R}^n$ we have:

$$\text{if} \quad x = \sum_{j=1}^{n} x_j v_j \quad \text{then} \quad Ax = \sum_{j=1}^{n} x_j \sigma_j u_j.$$

Proof. We use induction in m and n. We assume that the SVD exists for $(m-1)\times(n-1)$ matrices and prove it for $m \times n$. We assume $A \neq 0$; otherwise, we take $\Sigma = 0$ and U and V are arbitrary orthogonal matrices.

The basic step occurs when $n = 1$ (since $m \geq n$). We write

$$A = U\Sigma V^\top \quad \text{with} \quad U = \frac{A}{\|A\|}, \quad \Sigma = \|A\|, \quad V = 1,$$

where $\|\cdot\|$ is the 2-norm.

For the induction step, choose v so that

$$\|v\| = 1 \quad \text{and} \quad \|A\| = \|Av\| > 0.$$

Let

$$u = \frac{Av}{\|Av\|},$$

which is a unit vector. Choose $\tilde{U}$ and $\tilde{V}$ so that $U = [u, \tilde{U}]$ and $V = [v, \tilde{V}]$ are $m \times m$ and $n \times n$ orthogonal square matrices respectively. Now write

$$U^\top A V = \begin{bmatrix} u^\top \\ \tilde{U}^\top \end{bmatrix} \cdot A \cdot [v \ \tilde{V}] = \begin{bmatrix} u^\top Av & u^\top A\tilde{V} \\ \tilde{U}^\top Av & \tilde{U}^\top A\tilde{V} \end{bmatrix}.$$

Then

$$u^\top Av = \frac{(Av)^\top (Av)}{\|Av\|} = \|Av\| := \sigma$$

and

$$\tilde{U}^\top Av = \tilde{U}^\top u \|Av\| = 0.$$

We claim that $u^\top A\tilde{V} = 0$ too because otherwise

$$\sigma = \|A\| = \|U^\top AV\| \geq \|[1, 0, \ldots, 0]U^\top AV\| = \|[\sigma,\ u^\top A\tilde{V}]\| > \sigma,$$

a contradiction. Therefore,

$$U^\top AV = \begin{bmatrix} \sigma & 0 \\ 0 & \tilde{U}^\top A\tilde{V} \end{bmatrix} = \begin{bmatrix} u^\top Av & 0 \\ 0 & \tilde{A} \end{bmatrix}.$$

Now we apply the induction hypothesis that

$$\tilde{A} = U_1\Sigma_1 V_1^\top.$$

Hence,

$$U^\top AV = \begin{bmatrix} \sigma & 0 \\ 0 & U_1\Sigma_1 V_1^\top \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}\begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^\top$$

or

$$A = \left(U \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}\right)\begin{bmatrix} \sigma & 0 \\ 0 & \Sigma_1 \end{bmatrix}\left(V \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}\right)^\top,$$

which is our desired decomposition. □

The SVD has a large number of important algebraic and geometric properties, the most important of which are summarized in the following theorem.

Theorem 3. *Let $A = U\Sigma V^\top$ be the SVD of the $m \times n$ matrix A, $m \geq n$.*

(1) *Suppose A is symmetric and $A = U\Lambda U^\top$ be an eigendecomposition of A Then the SVD of A is $U\Sigma V^\top$ where $\sigma_i = |\lambda_i|$ and $v_i = u_i \mathrm{sign}(\lambda_i)$, where $\mathrm{sign}(0) = 1$.*

(2) *The eigenvalues of the symmetric matrix $A^\top A$ are σ_i^2. The right singular vectors v_i are the corresponding orthonormal eigenvectors.*

(3) *The eigenvectors of the symmetric matrix $AA^\top$ are σ_i^2 and $m - n$ zeroes. The left singular vectors u_i are the corresponding orthonormal eigenvectors for the eigenvalues σ_i^2. One can take any $m-n$ orthogonal vectors as eigenvectors for the eigenvalue 0.*

(4) *If A has full rank, the solution of*

$$\min_x \|Ax - b\| \quad \text{is} \quad x = V\Sigma^{-1}U^\top b.$$

(5)

$$\|A\|_2 = \sigma_1.$$

If A is square and nonsingular , then

$$\|A^{-1}\|_2 = \frac{1}{\sigma_n}.$$

(6) *Suppose*

$$\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0.$$

Then

$$\mathrm{rank}(A) = r,$$
$$\mathrm{null}(A) = \{x \in \mathbb{R}^n \ :\ Ax = 0 \in \mathbb{R}^m\} = \mathrm{span}(v_{r+1}, \ldots, v_n),$$

$$\operatorname{range}(A) = \operatorname{span}(u_1, \ldots, u_r).$$

(7)
$$A = U\Sigma V^\top = \sum_{i=1}^{n} \sigma_i u_i v_i^\top,$$

i.e., A is a sum of rank 1 matrices. Then a matrix of rank $k < n$ closest to A is

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\top, \quad \text{and} \quad \|A - A_k\| = \sigma_{k+1}.$$

5. Condition number

Let $f(x)$ be a vector-valued function that we need to evaluate. The condition number $\kappa(f;x)$ is the ratio of the relative error in f caused by the relative error in x provided that the change in x is small. Hence, we define κ as

(8)
$$\kappa(f;x) := \lim_{\epsilon\to 0} \max_{\|\Delta x\|=\epsilon} \frac{\|f(x+\Delta x) - f(x)\|/\|f(x)\|}{\|\Delta x\|/\|x\|}.$$

5.1. **Condition numbers for differentiable functions.** Let $f(x)$ be a differentiable vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$. Then

$$f(x+\Delta x) = f(x) + J(x+\theta\Delta x)\Delta x, \quad \text{where} \quad \theta \in (0,1),$$

and J is the Jacobian matrix of f with entries:

$$J_{ij}(x) := \frac{\partial f_i}{\partial x_j}.$$

Then

$$\kappa(f;x) = \lim_{\epsilon\to 0} \max_{\|\Delta x\|=\epsilon} \frac{\|x\|\|J(x+\theta\Delta x)\Delta x\|}{\|f(x)\|\|\Delta x\|}.$$

The maximum over $\Delta x \in \mathbb{R}^n$ such that $\|\Delta x\| = \epsilon$ is achieved if Δx is parallel to the first right singular vector v_1 of $J(x+\Delta x) = U\Sigma V^\top$. Therefore,

$$\kappa(f;x) = \frac{\|J\|\|x\|}{\|f(x)\|}.$$

5.2. **Condition number for matrix-vector multiplication.** A particular case is when $f(x)$ is a linear function, i.e., $f(x) = Ax$ where A is an $m \times n$ matrix. Then the Jacobian matrix of f is constant and is equal to A. Hence, the condition number for matrix-vector multiplication is

(9)
$$\kappa(A;x) = \frac{\|A\|\|x\|}{\|Ax\|} = \|A\|\frac{\|x\|}{\|Ax\|}.$$

Identity (9) shows that the condition number will be large if

$$\frac{\|Ax\|}{\|x\|} \ll \|A\|,$$

i.e., if there is a vector y that is elongated by A by much larger factor than x. Let us illustrate this phenomenon on a simple example from Bindel&Goodman (Chapter 4, page 89). Let

$$A = \begin{bmatrix} 1000 & 0 \\ 0 & 10 \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Then

$$Ax = \begin{bmatrix} 0 \\ 10 \end{bmatrix}.$$

Suppose x is perturbed by

$$\Delta x = \begin{bmatrix} \epsilon \\ 0 \end{bmatrix}. \quad \text{Then} \quad A(x + \Delta x) - Ax = A\Delta x = \begin{bmatrix} 1000\epsilon \\ 0 \end{bmatrix}.$$

The error in x is amplified by the factor of 1000 that is 100 times larger than the elongation of x. It is easy to check that for this example, $\kappa(A; x) = 100$.

5.3. **Condition number for solving linear system.** On the other hand, let us consider the problem of solving a linear system $Ax = b$, i.e., $f(x) = A^{-1}b$. We find:

$$\kappa(A^{-1}; b) = \|A^{-1}\| \frac{\|b\|}{\|A^{-1}b\|} = \|A^{-1}\| \frac{\|Ax\|}{\|x\|}. \tag{10}$$

The condition number for the linear system $Ax = b$ is large if some vector is stretched by A much less than the solution x (recall that $\|A^{-1}\| = 1/\sigma_n$, where σ is the smallest singular value of A).

What we often call the condition number of a matrix A defined as

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is the worst-case scenario condition number for either of the problems: matrix-vector multiplication and solving of linear system.

6. Basics of optimization problems

The most general optimization problem is

$$f(x) \to \min \quad \text{subject to} \quad x \in \Omega. \tag{11}$$

The function $f(x) : \mathbb{R}^n \to \mathbb{R}$ is called the *objective function* and the set Ω is the *constraint set*. Usually, Ω is defined by a collection of equations $c_i(x) = 0$, $i \in \mathcal{E}$, and inequalities $c_i(x) \geq 0$, $i \in \mathcal{I}$. A point in Ω is called *feasible*. If $\Omega \equiv \mathbb{R}^n$, the optimization problem in called *unconstrained*. Methods for solving unconstrained optimization problems are simpler than those for constrained ones. Methods for constrained optimization are built up upon theoretical results and methods for unconstrained optimization. The most pivotal pieces of theory for constrained optimization are the Karush-Kuhn-Tucker theory and duality. A very important textbook on 20th century optimization is J. Nocedal and S. Wright "Numerical Optimization" [10].

Even simple optimization problem does not necessarily have a solution. For example, the function $f(x)$ can be unbounded from below. Most optimization methods with objective

functions bounded from below are only guaranteed to find a stationary point x^* meaning $\nabla f(x^*) = 0$. This point is even not guaranteed to be a local minimizer!

There is a special class of problems called *convex* with convex function $f(x)$ and convex set Ω for which the solution exists and every local minimizer is also a global minimizer:

- the set Ω is convex if for any $x, y \in \Omega$, and any $0 < \alpha < 1$, $\alpha x + (1-\alpha)y \in \Omega$;
- the function $f(x)$ is convex on a convex set Ω if for any $x, y \in \Omega$, and any $0 < \alpha < 1$,

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y).$$

If this inequality is strict, $f(x)$ is *strictly convex*. In this case, the solution to problem (11) exists and is unique.

References

[1] W. E and B. Yu, "The deep ritz method: A deep learning-based numerical algorithm for solving variational problems," *Communications in Mathematics and Statistics*, vol. 6, pp. 1–12, 2018.

[2] Y. Khoo, J. Lu, and L. Ying, "Solving for high-dimensional committor functions using artificial neural networks," *Research in the Mathematical Sciences*, vol. 6, no. 1, p. 1, 2019.

[3] Q. Li, B. Lin, and W. Ren, "Computing committor functions for the study of rare events using deep learning," *Journal of Machine Learning Research*, vol. 151, p. 054112, 2019.

[4] Y. Wang, J. Ribeiro, and P. Tiwary, "Machine learning approaches for analyzing and enhancing molecular dynamics simulations," *Current Opinion in Structural Biology*, vol. 61, pp. 139–145, 2020.

[5] W. E, "The dawning of a new era in applied mathematics," *Notices of the Americal Mathematical Society*, vol. 68, no. 4, pp. 565–571, 2021.

[6] R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 5–30, 2006.

[7] B. Yu and E. Weinan, "The deep ritz method: A deep learning-based numerical algorithm for solving variational problems," *Commun. Math. Stat.*, vol. 6, pp. 1–12, 2018.

[8] I. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, pp. 2295–2317, 2011.

[9] J. W. Demmel, *Applied Numerical Linear Algebra*. SIAM, 1997.

[10] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2 ed., 2006.