Applications of Linear Algebra

Javier Peña
Carnegie Mellon University

Taylor Allderdice High School, Pittsburgh
April 23, 2025

Main take-away

Linear algebra (matrices, vectors, etc) is used everywhere.

To get a taste, we will see how matrices and vectors come up in:

- Image processing
- Networks
- Quantum computation
- Data science

Image processing

Puzzle

One of these pictures requires a lot less storage than the other one.

(Source: `skimage` Python library.)

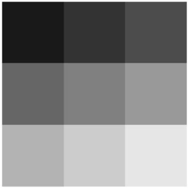
Can you guess which one?

Digital pictures = matrices

A digital black and white picture = matrix with entries in $[0, 1]$.

Example

$$\begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 \end{bmatrix} \quad \rightsquigarrow$$

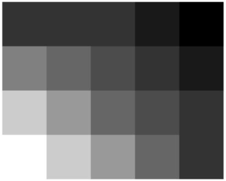


Intuition for image compression

How many numbers do we need to store the following pictures?

$$\begin{bmatrix} 0 & 0.25 & 0.5 & 0.75 \\ 0 & 0.25 & 0.5 & 0.75 \\ 0 & 0.25 & 0.5 & 0.75 \\ 0 & 0.25 & 0.5 & 0.75 \end{bmatrix} \rightsquigarrow$$

$$\begin{bmatrix} 0.25 & 0.2 & 0.15 & 0.1 & 0.05 \\ 0.5 & 0.4 & 0.3 & 0.2 & 0.1 \\ 0.75 & 0.6 & 0.45 & 0.3 & 0.15 \\ 1 & 0.8 & 0.6 & 0.4 & 0.2 \end{bmatrix} \rightsquigarrow$$



Low rank matrices

Notation:

- $\mathbb{R}^n = n \times 1$ column vectors
- $\mathbb{R}^{m \times n} = m \times n$ matrices with real entries.

Recall

Rank of $A \in \mathbb{R}^{m \times n}$ = number of linearly independent rows (or columns)

Rank-one matrix

If $A \in \mathbb{R}^{m \times n}$ has rank one, then $A = uv^\mathsf{T}$ for some $u \in \mathbb{R}^m, v \in \mathbb{R}^n$.

Low-rank matrix

If $A \in \mathbb{R}^{m \times n}$ has rank k, then

$$A = UV^\mathsf{T}$$

for some $U \in \mathbb{R}^{m \times k}, V \in \mathbb{R}^{n \times k}$.

Compression via singular value decomposition

Singular value decomposition (SVD)

Suppose $A \in \mathbb{R}^{m \times n}$. Then A can be factored as

$$A = UDV^\mathsf{T}$$

for some $U \in \mathbb{R}^{m \times m}, V \in \mathbb{R}^{n \times n}, D \in \mathbb{R}^{m \times n}$ such that

- D is diagonal with non-negative entries,
- $U^\mathsf{T}U = I_m$ and $V^\mathsf{T}V = I_n$.

Compressed version of A

Suppose the diagonal entries of D are $d_{11} \geq d_{22} \geq \cdots$.
Keep only the k largest ones, that is, approximate A with

$$V[:, :k]D[:k, :k]U[:, :k]^\mathsf{T}$$

Notation: $V[:, :k] =$ first k columns of V. Likewise for the others.

Connection between SVD and eigendecomposition

Suppose $A \in \mathbb{R}^{m \times n}$ with SVD $A = VDU^\mathsf{T}$.

The eigendecompositions of $A^\mathsf{T} A \in \mathbb{R}^{n \times n}$ and $AA^\mathsf{T} \in \mathbb{R}^{m \times m}$ are

$$A^\mathsf{T} A = V(D^\mathsf{T} D)V^\mathsf{T}$$

and

$$AA^\mathsf{T} = U(DD^\mathsf{T})U^\mathsf{T}.$$

Observe

$D^\mathsf{T} D \in \mathbb{R}^{n \times n}$ and $DD^\mathsf{T} \in \mathbb{R}^{m \times m}$ are diagonal with entries

$$d_{11}^2, d_{22}^2, \ldots.$$

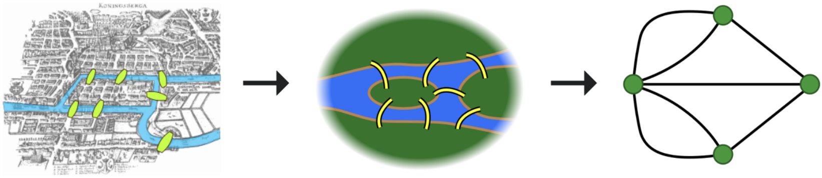
Those are the eigenvalues of $A^\mathsf{T} A$ and AA^T.

Networks

Graphs

Ancient problem: the seven bridges of Königsberg

Devise a walk through the city that would cross each of the seven bridges exactly once.

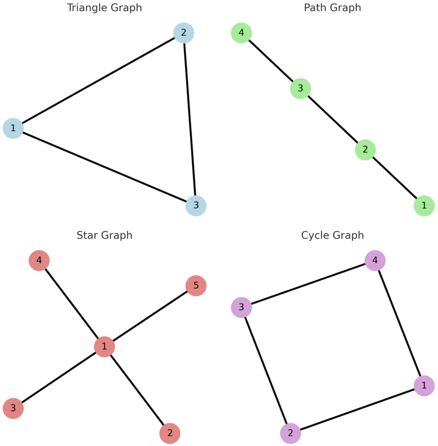


Euler solved this problem in 1735 and created graph theory.

Undirected graph (aka simple graph)

An undirected graph is a pair (V, E) where V is a finite set of vertices and E is a set of **unordered pairs** of elements of V.

Examples



Graphs and matrices

Suppose a graph (V, E) has vertices labeled $1, \ldots, n$.

Adjacency matrix

This is a $n \times n$ matrix A with entries $0, 1$ entries indicating the vertices joined by edges.

Adjacency matrix for the graphs on the previous slide:

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Graphs and matrices

Suppose a graph (V, E) has vertices labeled $1, \ldots, n$.

Laplacian matrix

$$L_{ij} = \begin{cases} \deg(i) & \text{if } i = j \\ -1 & \text{if } i, j \text{ are adjacent} \\ 0 & \text{otherwise.} \end{cases}$$

Laplacian matrix of the last two graphs on page 18:

$$\begin{bmatrix} 4 & -1 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Interesting fact

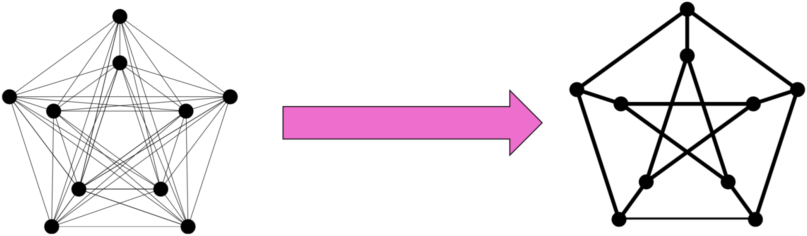
Number of connected components of G = dim of nullspace of L.

What else can we do with these matrices?

Short answer: a lot.

An interesting application of the Laplacian is *sparsification:* approximate a graph with a sparse one.

Example

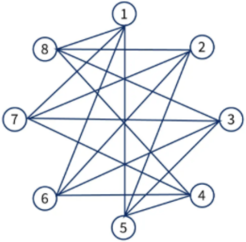


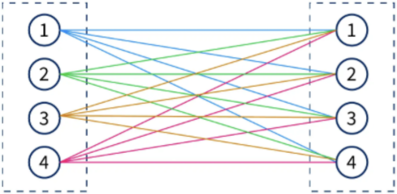
To sparsify a graph: use the eigendecomposition of its Laplacian.

Bipartite graphs

A graph $G = (V, E)$ is **bipartite** if we can partition $V = V_1 \cup V_2$ so that every edge connects a vertex in V_1 to a vertex in V_2.

Example



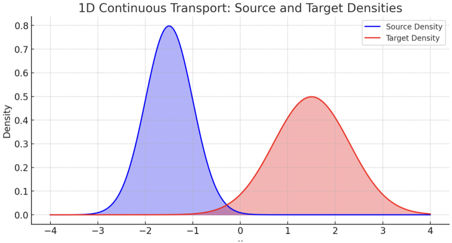


Can you relate this property to the incidence matrix?

Transport problem

Move some commodity from some sources to some destinations at minimum cost.

Continuous version: shift the mass from an initial distribution to a target distribution.



Formally introduced by French mathematician Monge in 1781.

Special case of transport problem

Suppose we have:

- n sources
- n destinations
- supply at each source = demand at each destination = 1

A transport plan is an $n \times n$ matrix $X \in \mathbb{R}^{n \times n}$ with the following properties:

- All entries of X are non-negative
- The entries of each row add up to 1
- The entries of each column add up to 1

That kind of matrix is called a **doubly stochastic** matrix.

Permutation matrices

Equivalent definitions of permutation matrix:

- Doubly stochastic matrix with exactly one entry equal to 1 in each row and each column.
- Matrix whose rows (or columns) are a permutation of the rows (or columns) of the identity matrix.

Examples

There are exactly six 3×3 permutation matrices:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

How many $n \times n$ permutation matrices are there?

Quantum computing

Picture from 1927 Solvay Conference on Physics

"The most intelligent picture ever taken"

Source: Wikipedia.

Bits and qubits

The **bit** is the fundamental concept of computation.

Quantum computation is built upon the analog concept of **qubit** (quantum bit).

A bit can be in one of two states: 0 or 1.

A qubit can be in the states $|0\rangle, |1\rangle$, or in a **superposition** of these two states:

$$\alpha|0\rangle + \beta|1\rangle$$

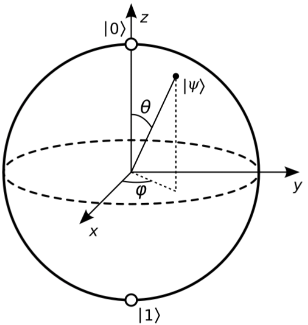
for $\alpha, \beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta|^2 = 1$.

Bloch sphere

Suppose the state of a qubit is $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.
Since $|\alpha|^2 + |\beta|^2 = 1$ we can rewrite $|\psi\rangle$ as

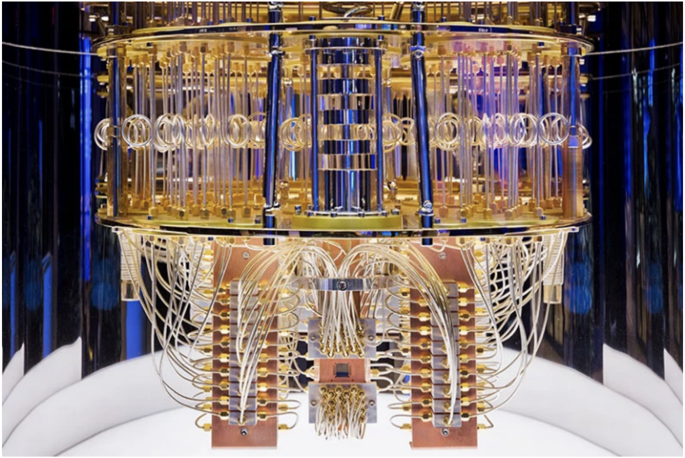
$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right)$$

for some $\theta \in [0, \pi], \phi, \gamma \in [0, 2\pi]$.
The state $|\psi\rangle$ can be visualized in the **Bloch sphere:**



Do quantum computers exist?

Yes. This is a picture of real quantum computer with 100 qubits:



Source

"First quantum computer to pack 100 qubits enters crowded race," Nature, Nov 2021.

Qubit operations

The state of a qubit can be seen as an element of $\mathbb{C}^2$:

$$\alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix} \in \mathbb{C}^2.$$

The analogue of bit operations, are qubit operations.

The classical NOT gate transforms bits: $0 \to 1$ and $1 \to 0$

Quantum NOT gate: $\alpha|0\rangle + \beta|1\rangle \to \beta|0\rangle + \alpha|1\rangle$.

Quantum NOT gate in matrix form:

$$X = \begin{bmatrix}0 & 1 \\ 1 & 0\end{bmatrix} \rightsquigarrow X \begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \begin{bmatrix}\beta \\ \alpha\end{bmatrix}$$

Other quantum gates

Hadamard gate ('square-root of NOT' gate):

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Pauli matrices

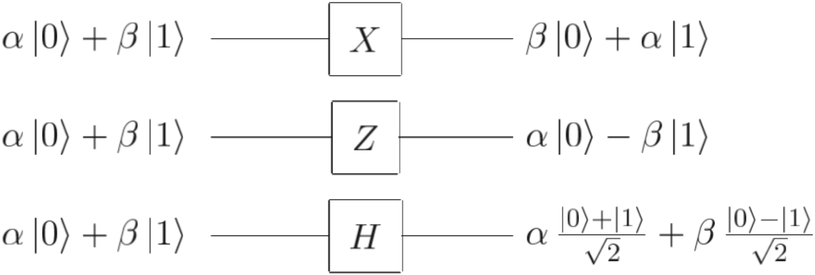
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The qubit operations defined by the above gates can be visualized in the Bloch sphere.

Observe

Each of H, X, Y, Z has eigenvalues -1 and 1.

Circuit representation of quantum gates

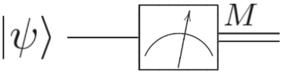


Measurement

Important difference between classical computation and quantum computation.

When we measure a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we get a probabilistic classical bit: 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

Circuit representation of measurement



Multiple qubits

State of two qubits:

$$\alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

with $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$.

Special two qubit state

The Bell state or EPR pair

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

This state is **entangled**: as soon as we measure the first qubit, we already know what the measurement of the second bit will be.

Quantum circuits

A quantum algorithm is a sequence of qubit operations.

A **quantum circuit** is a convenient way of describing a quantum algorithm. A quantum circuit has three main components:

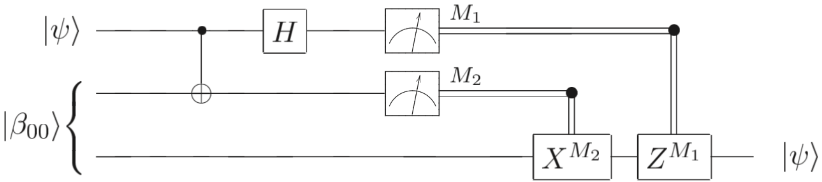
- Input qubits
- Quantum gates
- Wires
- Measurements

Quantum teleportation

Alice and Bob met long ago but now live far apart. When they met, they generated an EPR pair and each took one qubit of the EPR pair.

Many years later Alice wants to deliver a qubit $|\psi\rangle$ to Bob. How could she?

Answer: this can be accomplished by sending two classical bits.



Data science

Data

Numerical datasets are usually matrices.

Datasets with other kind of data (e.g., categorical data) can also be encoded as matrices.

In a dataset:

- Columns are usually "features"
- Rows are usually "observations".

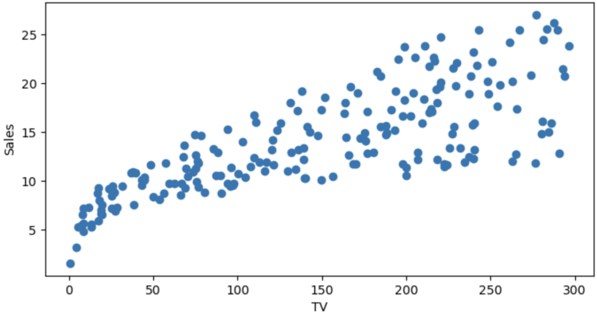
Example

Advertising dataset (source: Kaggle).

Features: TV, Radio, Newspaper, Sales

Linear regression

Scatter plot of Sales versus TV suggests a linear pattern:



Linear regression

Estimate a linear model

$$\text{Sales} \ = \beta_0 + \beta_1 \cdot \text{TV} + \epsilon.$$

How can we do that?

Least-squares procedure

Suppose we have n observations for two variables x, y and want to find the line that "best fits" the model

$$y = \beta_0 + \beta_1 x + \epsilon.$$

A popular approach is to minimize

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2.$$

Solution to least-squares problem

Construct the following vectors and matrices:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},\ X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix},\ \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}.$$

The least-squares problem can be written as follows:

$$\min_{\beta} (y - X\beta)^{\mathsf{T}} (y - X\beta)$$

Solution (with a bit of calculus):

$$X^{\mathsf{T}} X\beta - X^{\mathsf{T}} y = 0 \Rightarrow \beta = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y.$$

Multiple linear regression

Suppose we have k predictor variables $x^1, \ldots, x^k$ and a target variable y and want to find the line that "best fits" the model

$$y = \beta_0 + \beta_1 x^1 + \cdots + b_k x^k + \epsilon.$$

Construct the following vectors and matrices:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \ X = \begin{bmatrix} 1 & x_1^1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2^1 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & & & \\ 1 & x_n^1 & x_n^2 & \cdots & x_n^k \end{bmatrix}, \ \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}.$$

The least-squares problem can be written as follows:

$$\min_{\beta} (y - X\beta)^{\mathsf{T}} (y - X\beta)$$

Solution (with a bit of calculus):

$$X^{\mathsf{T}} X \beta - X^{\mathsf{T}} y = 0 \Rightarrow \beta = (X^{\mathsf{T}} X)^{-1} X^{\mathsf{T}} y.$$

Recall main take-away

Linear algebra (matrices and vectors) is used everywhere.

In particular, it is used extensively when we work with images, networks, quantum computation, and data.

Some references

- Szeliski, "Computer Vision: Algorithms and Applications"
- Ahuja, Magnanti, and Orlin, "Network Flows"
- Nielsen and Chuang, "Quantum Computation and Quantum Information"
- James, Witten, Hastie, and Tibshirani, "An Introduction to Statistical Learning"