Dimensionality Reduction and Linear Transformations

ECE 57000 — September 4, 2026

David I. Inouye

Wednesday established why dimensionality reduction is needed

We began with data that computers can process but people cannot inspect directly:

$$500 \text{ cells} \times 20{,}000 \text{ measurements per cell}$$

Goal 1: visualization

Reveal clusters, unusual points, and other hidden structure.

Goal 2: further analysis

Use fewer features for cheaper computation and potentially more reliable statistical estimation.

> **The unresolved question is what useful structure the reduced representation should retain.**

What kind of **structure** do we want to maintain?

Groups and neighborhoods. Keep nearby points close and distinct groups separate when those relationships matter.

Spread. Retain directions in which the observations vary rather than compressing meaningful variation.

Reconstruction. Prefer features that can recover points close to their original 2D locations when reconstruction matters.

How do we formalize these criteria?

We need the language of vectors, geometry, and linear transformations.

Formalization starts by naming one observation

For cell i, collect its d measured features into a vector:

$$\boldsymbol{x}_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{id} \end{bmatrix}^T \in \mathbb{R}^d$$

- i identifies the cell.
- j identifies a measured feature such as one gene.
- d is the number of observed features—not automatically the number of meaningful directions in the data.

A data matrix records features—not intrinsic dimension

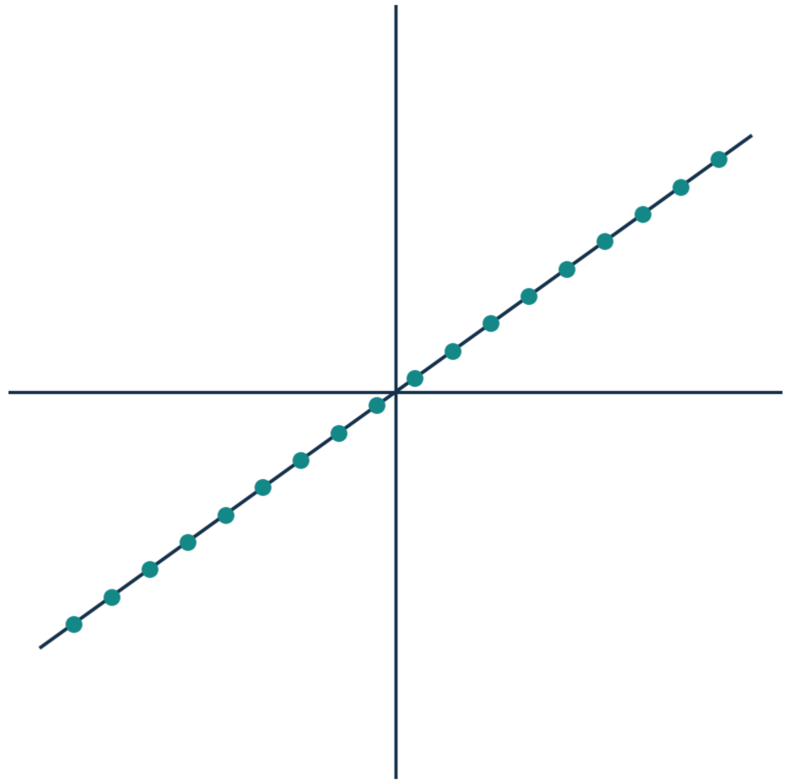
Stack n observations as rows:

$$X = \begin{bmatrix} - - - \boldsymbol{x}_1^T - - - \\ - - - \boldsymbol{x}_2^T - - - \\ \vdots \\ - - - \boldsymbol{x}_n^T - - - \end{bmatrix} \in \mathbb{R}^{n \times d}$$

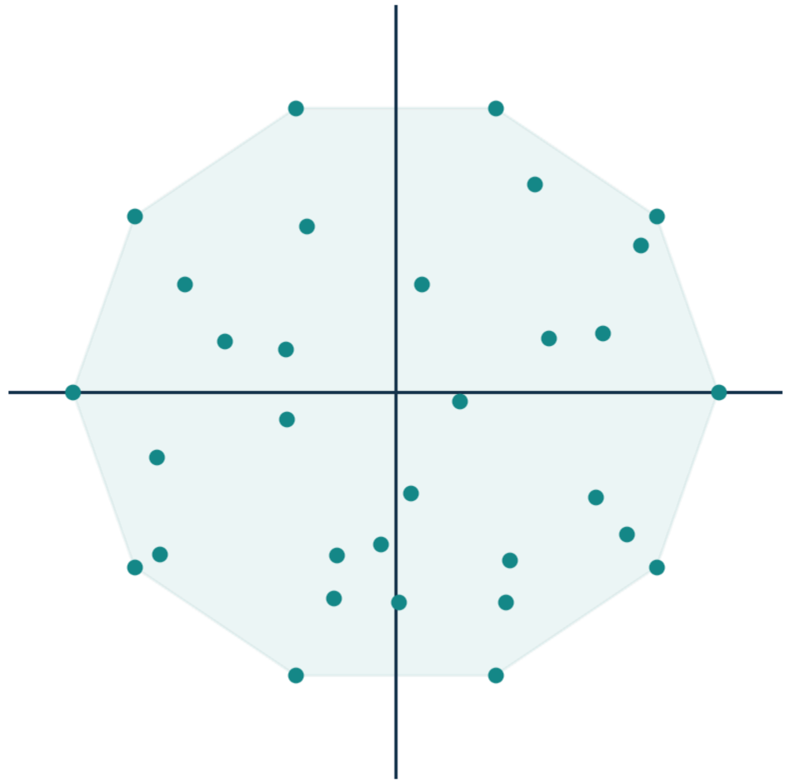
rows $=$ observations

columns $=$ measured features

1D structure in 2D



2D structure in 2D



Both datasets have two measured features; only one needs two independent directions to describe its variation.

We seek a function from observed to reduced features

For each observation, seek

$$f : \mathbb{R}^d \to \mathbb{R}^k, \qquad \boldsymbol{z}_i = f(\boldsymbol{x}_i), \qquad k \ll d,$$

where $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ contains the observed features and $\boldsymbol{z}_i \in \mathcal{Z} \subseteq \mathbb{R}^k$ contains the reduced features.

> **For now, assume f is linear. Later, we will learn nonlinear representation functions—but linear transformations are the foundation for analyzing both.**

To define, analyze, and choose this f, we need linear algebra for **geometry**, **transformations**, and **lower-dimensional structure**.

The problem organizes what comes next

> **We began with a need: make high-dimensional relationships inspectable without pretending every relationship survives.**

We now have:

1. an explicit purpose and candidate preservation criteria;
2. observations x_i, a data matrix X, and desired reduced features Z;
3. a visual distinction among neighborhoods, spread, and reconstruction; and
4. a concrete reason to develop the linear algebra needed to finish the model.

Linear Transformations: What Do They Do?

A linear representation turns the problem into a matrix question

We began with a representation function:

$$f : \mathbb{R}^d \to \mathbb{R}^m, \qquad \boldsymbol{y} = f(\boldsymbol{x}).$$

For this first representation model, assume f is linear:

$$\boldsymbol{y} = A\boldsymbol{x}, \qquad A \in \mathbb{R}^{m \times d}.$$

What can this transformation do to a high-dimensional feature space?

Matrix multiplication generalizes scalar multiplication

One dimension

$$y = ax$$

- x, y, a are scalars.
- a scales one number line.
- The sign can reverse direction.

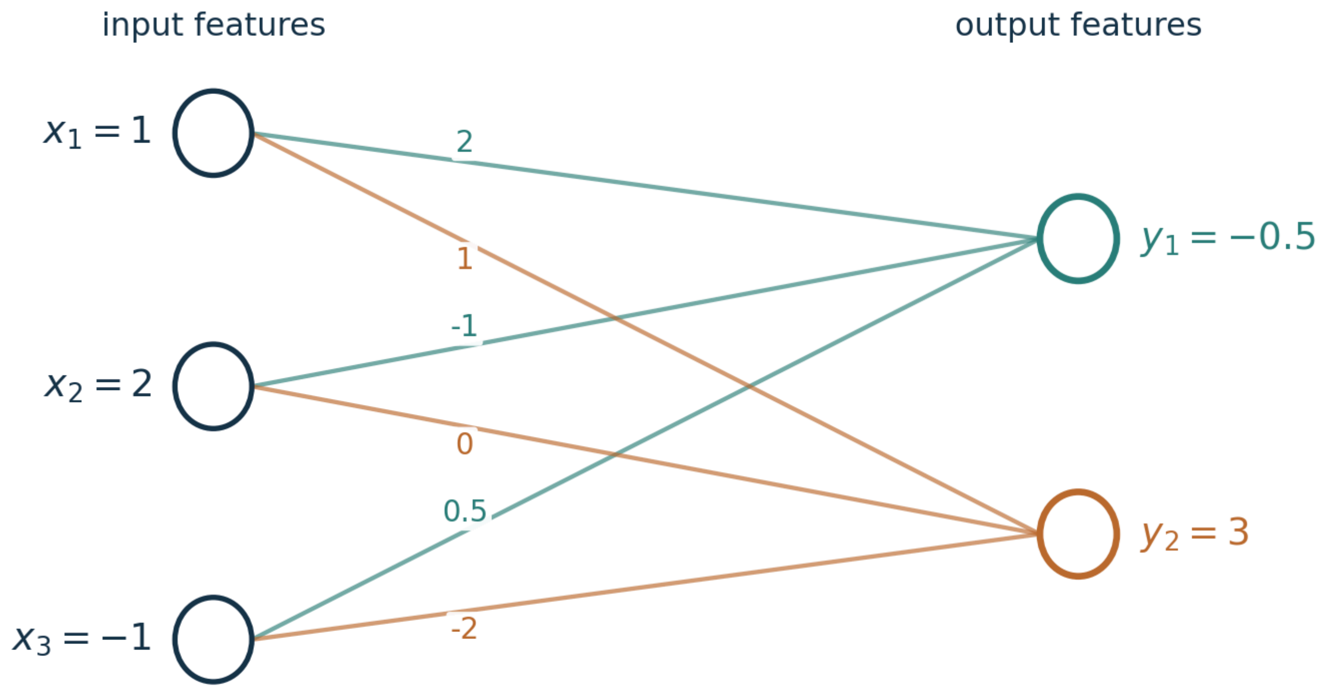
Higher dimensions

$$\boldsymbol{y} = A\boldsymbol{x}$$

- $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors.
- A can scale different directions differently.
- Input and output can have different dimensions.

a becomes A: one scalar action becomes a transformation of a feature space.

Each output feature receives a weighted connection from every input



$$A = \begin{bmatrix} 2 & -1 & 0.5 \\ 1 & 0 & -2 \end{bmatrix}$$

- A teal edge contributes to y_1.
- An orange edge contributes to y_2.
- Every edge weight is one entry a_{ji}.

Every output is a different linear combination of the inputs

$$\begin{aligned} y_1 &= 2x_1 - x_2 + 0.5x_3 \\ &= 2(1) - 2 + 0.5(-1) \\ &= -0.5 \end{aligned}$$

$$\begin{aligned} y_2 &= x_1 + 0x_2 - 2x_3 \\ &= 1 + 0 - 2(-1) \\ &= 3 \end{aligned}$$

$$\underbrace{\begin{bmatrix} -0.5 \\ 3 \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} 2 & -1 & 0.5 \\ 1 & 0 & -2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}}_{x}$$

One matrix applies the same transformation to every observation

For observation i:

$$\boldsymbol{y}_i = A\boldsymbol{x}_i, \qquad \boldsymbol{x}_i \in \mathbb{R}^d, \quad \boldsymbol{y}_i \in \mathbb{R}^m.$$

The same entries of A define every output feature for every observation.

- **Convention:** Vectors such as $\boldsymbol{x}_i$ are usually written as columns.
- **Transpose:** A^T swaps rows and columns: $A \in \mathbb{R}^{m \times d} \Rightarrow A^T \in \mathbb{R}^{d \times m}$.

One linear operator gives a consistent representation rule for the entire dataset.

Which batch equation has the right shape?

Observations are rows:

$$X = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 2 \\ 2 & -1 & 1 \end{bmatrix}$$

Use the same transformation:

$$A = \begin{bmatrix} 2 & -1 & 0.5 \\ 1 & 0 & -2 \end{bmatrix}$$

Compute Y with one matrix multiplication

$$Y = ???$$

$$\underbrace{Y}_{3\times 2} = \underbrace{X}_{3\times 3}\,\underbrace{A^T}_{3\times 2} = \begin{bmatrix} -0.5 & 3 \\ 0 & -4 \\ 5.5 & 0 \end{bmatrix}$$

Each row of Y is $(A\boldsymbol{x}_i)^T$.

Matrix algebra preserves some scalar rules and rejects others

Valid for scalars and matrices

$$A(B\boldsymbol{x}) = (AB)\boldsymbol{x}$$

$$A(\boldsymbol{x} + \boldsymbol{z}) = A\boldsymbol{x} + A\boldsymbol{z}$$

$$AI = A$$

Order matters for matrices

$$AB \neq BA \quad \text{in general}$$

$$(AB)^T = B^T A^T$$

Changing the order can change the value or make the product undefined.

Dimensions expose many invalid manipulations

Let $A \in \mathbb{R}^{m \times d}$ and $B \in \mathbb{R}^{d \times p}$.

Valid

$$AB \in \mathbb{R}^{m \times p}$$

$$(AB)^T = B^T A^T$$

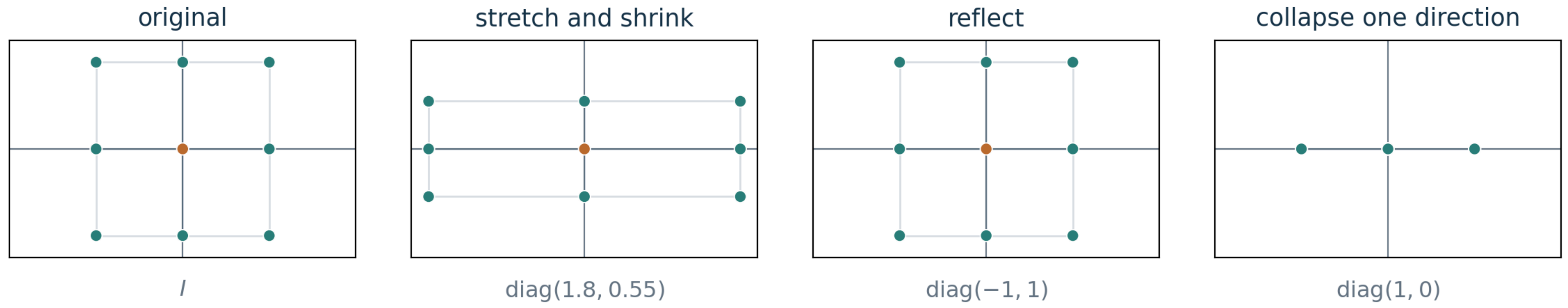
Not guaranteed to exist

$$BA$$

The inner dimensions would require $p = m$.

> **Shape is part of the mathematical claim, not an implementation detail.**

A diagonal matrix applies a separate 1D scaling to each feature



A zero diagonal entry behaves like $a = 0$: one direction becomes a **partial zero**.

The Euclidean norm measures size and distance

For $\boldsymbol{x} \in \mathbb{R}^d$:

$$\|\boldsymbol{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_d^2}$$

For two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$\operatorname{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$$

The subscript 2 identifies the Euclidean norm. Other norms measure size differently.