SVD Interlude

A Geometric Buildup to the SVD

From Input/Output Spaces to Orthogonal Directions That Stay Orthogonal

Big Idea

Often a matrix represents a process that takes in one kind of data and produces another.

Examples:

- 3D position $\to$ 2D screen coordinates

. . .

- RGB color (3 numbers) $\to$ grayscale brightness (1 number)

. . .

- 3 forces on an object $\to$ net torque ($3 \to 1$)

. . .

- 5 features $\to$ a prediction score

So a matrix can represent a transformation

$$A : \mathbb{R}^n \to \mathbb{R}^m$$

The input space and output space may be **different worlds**.

. . .

A Concrete Example (3 → 2)

Consider

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

Then

$$A(x, y, z) = (x, y).$$

. . .

Geometrically:

- the component of the input in the x direction shows up in the output
- the component of the input in the y direction shows up in the output
- the component of the input in the z direction is completely ignored

. . .

So:

- the entire xy-plane survives
- everything in the z direction gets squashed to zero

Already we see:

> Some directions in the input space matter, and some directions don't.

. . .

Decomposing Any Vector

Every vector $\mathbf{x} \in \mathbb{R}^3$ can be written as

$$\mathbf{x} = (\text{part in the } xy\text{-plane}) + (\text{part in the } z\text{-direction}).$$

Apply A:

$$A\mathbf{x} = A(\text{xy part}) + A(\text{z part}).$$

But the second term is always zero, so

$$A\mathbf{x} = A(\text{xy part}).$$

> Before doing anything interesting, a transformation throws away the part of the input it can't see.

. . .

Example: Rank 1

Sometimes a transformation effectively responds to only one input direction.

For instance, let

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} : \mathbb{R}^2 \to \mathbb{R}^3.$$

What It Does

It takes (x, y) and returns

$$A(x, y) = (x + y, 0, 0).$$

Multiply It Out

Multiply A by a vector:

$$A \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x + y \\ 0 \\ 0 \end{bmatrix}$$

. . .

Notice that the output depends only on the combination

$$x + y.$$

So the component of the input in the direction

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

affects the output, but the component of the input in the orthogonal direction

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

is completely ignored.

So even though the input space is 2-dimensional, the transformation really only responds to the component of the input in **one direction**.

Now:

- all outputs lie along a **line** in $\mathbb{R}^3$
- the map only uses **one** independent input direction

So

$$\operatorname{rank}(A) = 1.$$

Geometrically:

- the transformation responds to the component of the input in only one direction
- all outputs are constrained to a one-dimensional slice of the output space.

This will show up later as having only **one nonzero singular value**.

. . .

Takeaway

The number of independent directions the map can actually do anything with is its **rank** — regardless of input or output dimension.

From Rank to SVD

Rank tells us *how many* directions a linear map can act on. SVD tells us *which* directions, and *how strongly*.

The SVD Motivation

Let $r = \text{rank}(A)$. We'll try to pick orthonormal input directions $\mathbf{v}_1, \ldots, \mathbf{v}_r$ that capture everything A can actually do.

Ask:

> Can we choose input directions $\mathbf{v}_1, \ldots, \mathbf{v}_r$ so that their images $[A\mathbf{v}_1, \ldots, A\mathbf{v}_r]$ are **orthogonal** in the output space?

That would mean the transformation treats these directions **independently**.

Define

$$\mathbf{w}_j = A\mathbf{v}_j.$$

We want:

$$\mathbf{w}_i^T \mathbf{w}_j = 0 \quad (i \neq j).$$

But

$$\mathbf{w}_i^T \mathbf{w}_j = (A\mathbf{v}_i)^T (A\mathbf{v}_j) = \mathbf{v}_i^T A^T A \mathbf{v}_j.$$

$$\mathbf{w}_i^T \mathbf{w}_j = (A\mathbf{v}_i)^T (A\mathbf{v}_j) = \mathbf{v}_i^T A^T A \mathbf{v}_j.$$

The output vectors will be orthogonal if we choose the $\mathbf{v}_j$ to be eigenvectors of $A^T A$.

Let

$$A^T A \mathbf{v}_j = \sigma_j^2 \mathbf{v}_j.$$

Then

$$\|\mathbf{w}_j\|^2 = (A\mathbf{v}_j)^T (A\mathbf{v}_j) = \mathbf{v}_j^T A^T A \mathbf{v}_j = \sigma_j^2.$$

So the outputs are orthogonal, with known lengths.

. . .

Normalize the Outputs

Define

$$\mathbf{u}_j = \frac{1}{\sigma_j}\mathbf{w}_j = \frac{1}{\sigma_j}A\mathbf{v}_j.$$

Then

$$\mathbf{u}_1, \dots, \mathbf{u}_r \text{ are orthonormal}$$

and

$$A\mathbf{v}_j = \sigma_j \mathbf{u}_j.$$

. . .

Matrix Form

Let

$$V_r = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_r], \qquad U_r = [\mathbf{u}_1 \ \cdots \ \mathbf{u}_r], \qquad \Sigma_r = \operatorname{diag}(\sigma_1, \dots, \sigma_r).$$

Then the relations $A\mathbf{v}_j = \sigma_j \mathbf{u}_j$ become

$$AV_r = U_r \Sigma_r.$$

. . .

Geometric Interpretation

For any input $\mathbf{x}$:

1. express $\mathbf{x}$ in the orthonormal directions $\mathbf{v}_j$
2. scale each coordinate by σ_j
3. rebuild the result in the orthonormal directions $\mathbf{u}_j$

So, in the right orthonormal coordinates in the input and output spaces,

> Multiplication by A is just independent scalings of the components along orthogonal directions in the two spaces.

. . .

Extending to Full Bases

We've described what happens to the r independent directions in the input space.

. . .

But what if the input space has more than r dimensions? Or the output space?

. . .

The trick is to extend the orthonormal bases to full bases.

- Extend $\mathbf{v}_1, \dots, \mathbf{v}_r$ to an orthonormal basis of $\mathbb{R}^n$
- Extend $\mathbf{u}_1, \dots, \mathbf{u}_r$ to an orthonormal basis of $\mathbb{R}^m$

Let's start with the case where the input space has more than r dimensions.

. . .

We started with the r singular vectors $\mathbf{v}_1, \dots, \mathbf{v}_r$. These capture everything A can actually do.

. . .

The rest of the input space is the null space of A, and our additional vectors will be a basis for this null space.

. . .

Let's call these additional vectors $\mathbf{v}_{r+1}, \dots, \mathbf{v}_n$. We know that

$$A\mathbf{v}_{r+1} = \mathbf{0}, \quad \dots, \quad A\mathbf{v}_n = \mathbf{0}.$$

Similarly, we can extend the singular vectors $\mathbf{u}_1, \dots, \mathbf{u}_r$ to a basis for the output space.

. . .

Let's call these additional vectors $\mathbf{u}_{r+1}, \dots, \mathbf{u}_m$. They represent directions that cannot be reached by $A\mathbf{x}$ for any $\mathbf{x}$.

. . .

This means that the projection of any output $A\mathbf{x}$ onto any of these additional vectors must be zero:

$$\mathbf{u}_{r+k}^T A\mathbf{x} = 0 \text{ for all } \mathbf{x}.$$

for $k = 1, \dots, m - r$.

. . .

And the only way this can be true for all $\mathbf{x}$ is if

$$\mathbf{u}_{r+k}^T A = \mathbf{0}$$

The transformation in adapted coordinates

We have built orthonormal bases:

For the **input space**:

$$V = [\mathbf{v}_1 \ \cdots \ \mathbf{v}_r \ \mathbf{v}_{r+1} \ \cdots \ \mathbf{v}_n].$$

For the **output space**:

$$U = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_r & \mathbf{u}_{r+1} & \cdots & \mathbf{u}_m \end{bmatrix}.$$

Split them into blocks:

$$V = \begin{bmatrix} V_r & V_0 \end{bmatrix}, \qquad U = \begin{bmatrix} U_r & U_0 \end{bmatrix}.$$

where

- V_r: directions where something happens
- V_0: null space directions
- U_r: directions actually reached
- U_0: directions never reached

Put A between the two bases:

Consider the matrix

$$U^T AV.$$

$$\begin{bmatrix} U_r^T \\ U_0^T \end{bmatrix} A \begin{bmatrix} V_r & V_0 \end{bmatrix} = \begin{bmatrix} U_r^T AV_r & U_r^T AV_0 \\ U_0^T AV_r & U_0^T AV_0 \end{bmatrix}$$

$$\begin{bmatrix} U_r^T AV_r & U_r^T AV_0 \\ U_0^T AV_r & U_0^T AV_0 \end{bmatrix}$$

Each block has a geometric meaning.

Top-right block: null space dies

Because

$$AV_0 = 0,$$

we get

$$U_r^T AV_0 = 0.$$
