$\perp$ → orthogonal / Perpendicular,

$\perp\!\!\!\perp$ ⇒ probabilistic independence

(14) Matrix Multiplication

$$\left\{ \begin{array}{l} A \in R^{m \times n}, \quad B \in R^{n \times p} \\ C = AB \in R^{m \times p} \end{array} \right\}$$

If A transforms first, then B. combined is AB. Real use of this in ML

- Linear Regression
- Least Squares
- Linear layers
- Solving Normal Equations
- Identifiability of Parameters

Linear algebra is not about solving, it also about understanding when exact solving is impossible

Linear Map $x \mapsto Ax$ ⇒ $Ax = b$ (Nothing but multiplication)

This is how its read in AI/ML

Means this

©Smiley

ASK WHEN TO DO MULTIPLICATION

What questions we have to ask?

⇒
- What output can it produce?
- Can it produce a given target?
- If yes, is the input unique?
- If not, why not?
- What directions get lost?
- Why does this become regression, least squares, feature redundancy and dimensionality reduction in ML?

Matrix Multiplication is for Linear equations.
Not like $x^2+y^2+z^2=1$ ⟶ sphere is not linear equation.

Basic rule ⇒ $A \in R^{m \times n}$, $x \in R^n$

$$Ax \in R^m$$

↳ how multiplication happens.

Matrix multiplication is thought as of columns, because it ease higher dimension

{Why columns are important ⇒}

If $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

$a_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$, $a_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$

So it feels like :

$$A = [a_1 \; a_2 \; a_3 \; \ldots\ldots \; a_n]$$

a_1 = first Column , a_2 = second Coulmn

$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ then, $\boxed{Ax = a_1x_1 + a_2x_2 + \cdots + a_nx_n}$

e.g. $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $x \begin{bmatrix} 5 \\ 6 \end{bmatrix}$

$a_1 = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$, $a_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$

$$Ax = 5a_1 + 6a_2$$

$$Ax = 5\begin{bmatrix} 1 \\ 3 \end{bmatrix} + 6\begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 5 \\ 15 \end{bmatrix} + \begin{bmatrix} 12 \\ 24 \end{bmatrix} = \begin{bmatrix} 17 \\ 39 \end{bmatrix}$$

15 **Image** ⇒ all outputs that this matrix can produce

$\text{Im}(A)$ = all reachable outputs

or $\boxed{\text{im}(A)}$

16 **Column Space** ⇒ all vectors you can make by combining the columns of the matrix

$\text{Col}(A)$ = all combinations of the columns

[It uses span to create column space]

17 **Span** ⇒ If you have some vectors $v_1, v_2, v_3 \cdots\cdots v_k$ then

$$\text{Span}(v_1, \cdots\cdots v_k)$$

means, ⇒ all linear combinations of these vectors.

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots\cdots + \alpha_k v_k$$

for all choices of numbers $\alpha_1, \alpha_2 \cdots\cdots, \alpha_k$

e.g. ⇒ • if you are given some arrows.

- You can stretch them, shrink them and add them
- All place you can reach by doing that = Span

R^2 = 2D Coordinate world
R^3 = 3D Coordinate world
R^n = higher-dimensional Coordinate world

SPAN ⇐ Deep Dive

→ One vector (2-D) spans a line through origin
→ Two vector (2-D), independent, spans entire space
→ Three independent vector (3D), spans entire 3-D space

Independent vector

$$v_1 \neq c v_2$$

$v = \begin{bmatrix}1\\2\end{bmatrix}$, then, $\text{span}\{v\} = \{cv : c \in R\} = \left\{\begin{bmatrix}c\\2c\end{bmatrix} : c \in R\right\}$

⇑

1-D ($\begin{bmatrix}c\\2c\end{bmatrix}$, can move on one line only)

$$v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad V_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$a v_1 + b v_2 = \begin{bmatrix} a \\ b \end{bmatrix}$$

Now a, b can be anything, so you can reach every point in 2D

but $\begin{bmatrix} c \\ 2c \end{bmatrix}$, previous page can move only on line, so

- one independent line / vector spans a line
- two independent vectors span the whole plane

$\boxed{\text{IMAGE}}$ ⇒ Deep-Dive

$$\text{Im}(A) = \{Ax : x \in \mathbb{R}^n\}$$

Image = all outputs the matrix can produce

- Imagine feeding every possible input vector into the matrix.
- Collect every output you ever get
- The full set is the image
- Image is set, not one single value.

$\Rightarrow$ Span $\Rightarrow$ "What can I build from these arrows?"
$\Rightarrow$ Image $\Rightarrow$ "where can this machine send points?"

$\Rightarrow$ Span $\Rightarrow$ Generated set from choosen vectors
$\Rightarrow$ Image $\Rightarrow$ reachable output set of a map

Is Span = Image, always?

$\Downarrow$

Linear Algebra, Yes

But,

$$f: R \to R, \quad f(x) = x^2$$

Image is $= [0, \infty)$, because x^2

Span $\Rightarrow (-\infty, -\infty)$, input $\Leftarrow$

Difference

Confusing Symbols

$\varnothing$ = empty set $\emptyset$ = empty set

Φ = usually capital Greek Phi, not emty set

$\backslash\varnothing$ = non standard emty set

$$\boxed{\{x \in \mathbb{R} : x^2+1=0\} = \varnothing}$$

example of emty set

$\in$ = inside as an element, $x \in A$
$\subseteq$ = entirely contained, $\{x\} \subseteq A$

$\cup$ = Combine
$\cap$ = Common
$\varnothing$ = nothing unside

Important Symbols

COLUMN SPACE ⇒ Deep Dive

Matrix A, has columns $a_1, a_2 \cdots a_n$

$$Col(A) = span(a_1, \cdots a_n)$$

⇒ Column Space = all vectors you can build from the columns.

(18) $Ker(A) = \{x : Ax = 0\}$

⇒ all inputs vectors that produce zero output.

- Kernel tells you which iput directions the matrix completely ignores.
- because if $v \in ker(A)$, then:

$$Av = 0$$

So if we move/chang v then also output does not change.

$$\boxed{\text{So, } A(x+v) = Ax + Av = Ax}$$

kernel is,

- invisible direction
- lost direction
- undetectable movement in input space.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad Ax = \begin{bmatrix} x_1 \\ 0 \end{bmatrix}$$

To be in kernel, need:

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

So $x_1 = 0$, x_2 can be anything

hence, $\ker(A) = \left\{ \begin{bmatrix} 0 \\ t \end{bmatrix} : t \in \mathbb{R} \right\}$

kernel named on word kill

- It measures space which does not change anything
- Hence kernel is often called null space

Suppose some transformation crushes some direction completely. Those crushed directions form the kernel

eg. $T(x, y) = (x, 0)$

This map kills horizonal part, but kills vertical part

Now

$$T(x, y) = (0, 0)$$

$$\text{Ker}(T) = \left\{ \begin{bmatrix} 0 \\ y \end{bmatrix} : y \in \mathbb{R} \right\}$$

$\left\{ \begin{array}{l} \text{image} = \text{what survives as output} \\ \text{kernel} = \text{what gets annihilated} \end{array} \right\}$

$$\boxed{\text{Ker}(A) = \text{Null}(A)}$$

|| SOLVING - USING - MATRIX - ELIMINATION ||

$$\begin{cases} x_1 + 2x_2 = 5 & -① \\ 3x_1 + 4x_2 = 11 & -② \end{cases}$$

Augmented Matrix

$$\left[\begin{array}{cc|c} 1 & 2 & 5 \\ 3 & 4 & 11 \end{array}\right] \Rightarrow \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array}\right]$$

Steps how we reached here

$R_2 \leftarrow R_2 - 3R_1$ -①

$R_2 \leftarrow -\frac{1}{2} R_2$ -②

$R_1 \leftarrow R_1 - 2R_2$

STEPS

If $Ax = b$

RULES

- A = Coefficient matrix
- x = unknown matrix
- b = right-side vector
- $[A|b]$ = Augmented matrix

Terms

$$\boxed{\begin{array}{l}\text{Homogeneous system} \Rightarrow Ax = 0 \\ \text{Non-Homogeneous system} \Rightarrow Ax = b\end{array}}$$

Homogeneous is about reaching zero output
Non-Homogeneous is about reaching target output.

How matrix elimination method can be thought of:

$$x_1 + 2x_2 = 5 \Rightarrow R_1$$
$$3x_1 + 4x_2 = 11 \Rightarrow R_2$$

These are two lines, legal operations

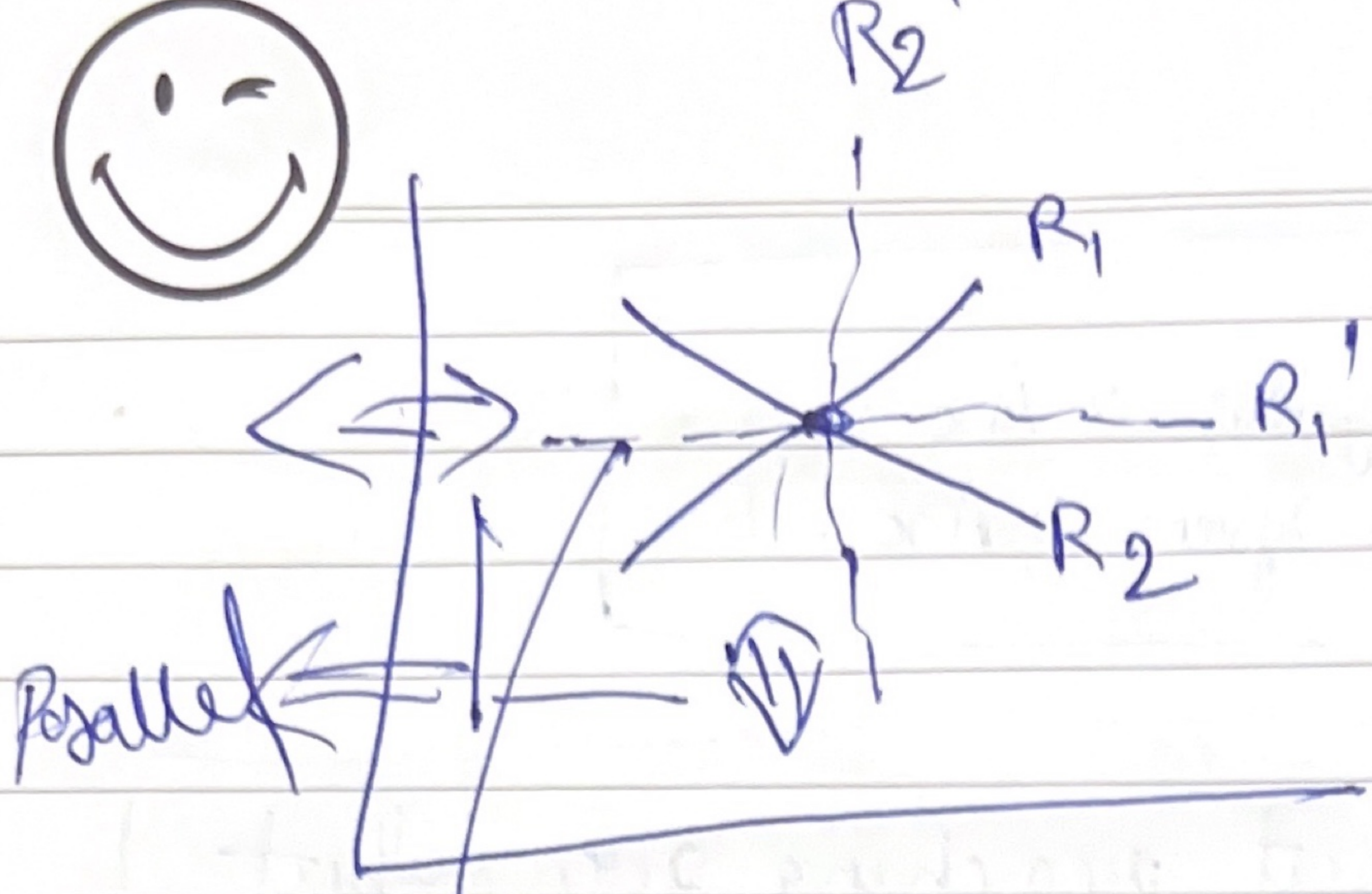
3-Rule-set

- swap rows
- multiple row by non scalar
- add multiple of one row to another

$$\boxed{R_2 \leftarrow R_2 - 3R_1}$$

This is allowed as $R_2 - 3R_1$ is new line passing through same intersection point.

©Smiley



If we find two lines ||al to x axis & y-axis.

~~and skip these others to left & down.~~

Then Matrix becomes identy matrix. and

$I x = b$

then, $x = b$

So, point to understand when,

$\boxed{x = 1, y = 2}$ → they are still two lines.

which intersect on (1, 2). Hence to solve we make $A \to I$ and finally we obtain (x, y).

Now,

$$Ax = b$$

Where we made $A \to I$, so whatever will be x, will be b. So

Image ⇒ Whole plane
Kernel ⇒ only Zero
Rank ⇒ 2

Meaning, matrix reaches everything in 2D and looses nothing (because kernel is only zero)

(19) $\text{rank}(A) = \dim(\text{col}(A))$

→ dimension of column space

→ it tells you how many output direction survive

Kernel tells what gets crushed to zero.
no solution happens when target b lies outside column space

→ number of independent output directions

Now,

$Ax = b$ has a solution $\iff$ $b \in Col(A)$

So;

- if b is an image/column space → at least one solution
- if b is outside image/column space → no solution

Then among solvable cases;

- if kernel is only $\{0\}$ → one solution
- if kernel has non-zero vectors → many solutions

Existence depends on Image
Uniqueness depends on Kernel

Why kernel decide one vs many solution

$Ax_0 = b$, if $Av = 0$ (when $v \neq 0$)

then $A(x_0 + v) = b$ (means second solution)