# LINEAR ALGEBRA in Data Science and AI

Shameek Bhattacharjee (Asst. Prof. WMU, Dept. of CS)

Scribe: Shourav Das (UG Student, WMU, Dept. of CS)

This material is part of NSF grant OAC-2017389

# Vectors in Computer Science

Physics → Magnitude and direction

Computer Science → An array of numbers (an **ordered** list of numbers)

Mathematics →identifying points in space, each element of a vector gives the coordinate along one axis.

Vectors in CS is represented by lower case bold symbol say **v**

$$\mathbf{v} = [v_1 \ldots v_i \cdots v_n]$$

**v** represents **a single point** in a cartesian coordinate **system of n-dimensions. Mathematically** → $v_i \in R^n$

(each element is in R, and the vector has n elements)

Each point can be represented as a vector → an arrow connecting origin (the tail) to the point (the tip)

In this figure (a), $v^{(1)}$ is a vector in the an n=2 dimensional coordinate system and in figure (b) it is in an n=3 dimensional coordinate system

Convention: In linear algebra, all vectors tail is fixed to the origin of this coordinate system (different from physics)

Length of the arrow = *magnitude* of the vector $v^{(1)}$

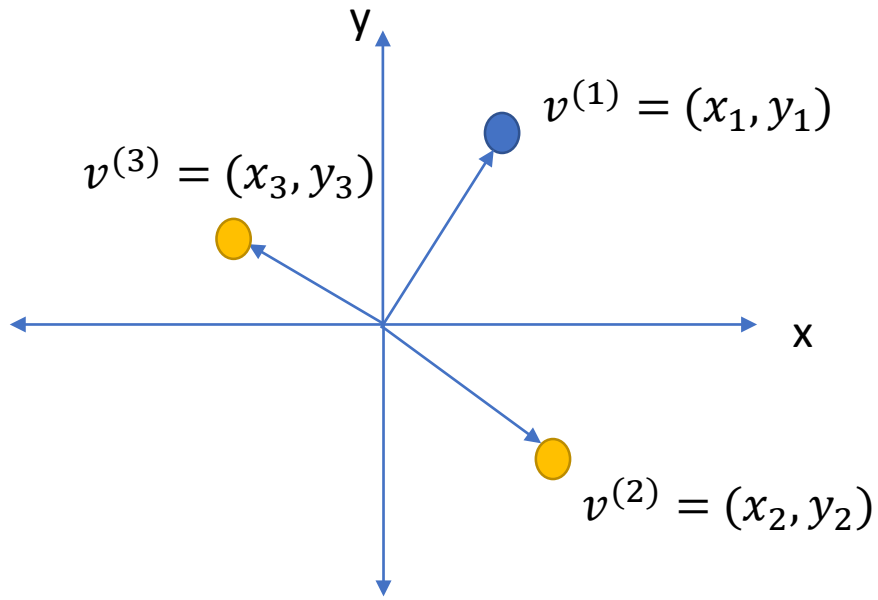



Figure : a

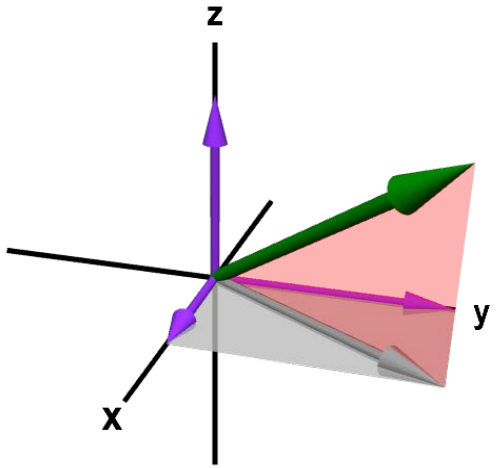
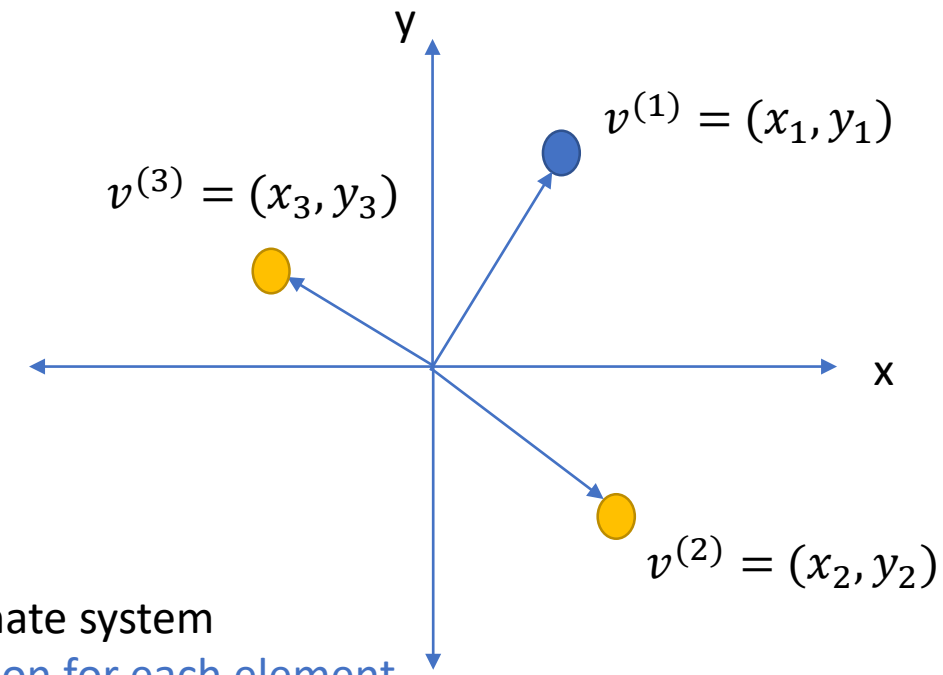



Figure: b

# Linear algebra in data science/AI/ML

| Training example # | Feature #1 | Feature #2 |
|---|---|---|
| 1 | $x_1$ | $y_1$ |
| 2 | $x_2$ | $y_2$ |
| 3 | $x_3$ | $y_3$ |

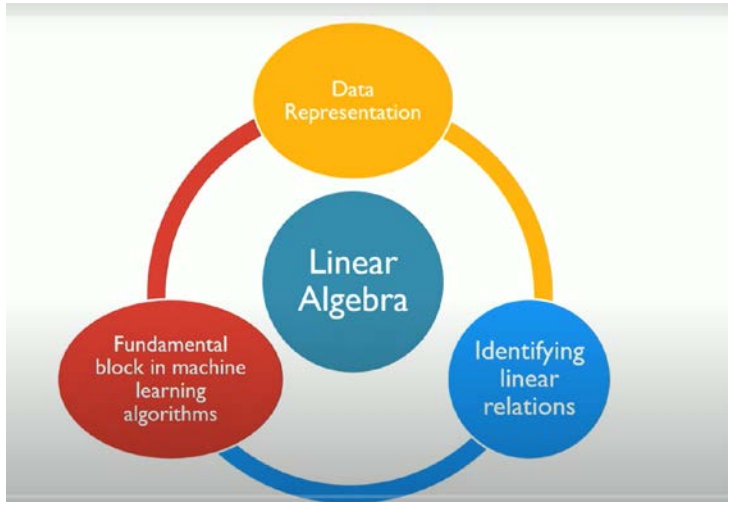



A huge dataset with ***n*** number of features
can be represented as points in an n-dimensional cartesian coordinate system
A vector of n elements is an n-dimensional vector with one dimension for each element.
(i) Visual intuition
(ii) other mathematical convenience (we will see later)

Conclusion: the coordinate axes could be equivalent to
Attributes/features/covariates/regressors/independent variables

The number of vectors you get is equal to the number of training data instances

# Linear algebra in data science/AI/ML




Vectors do not just represent data. They also help represent our model. Many types of Machine Learning models represent their learning as vectors. All types of neural networks do this. Given some data, it will learn dense representations of that data. These representations are essentially categories kin to recognize new given data.

```
1   import numpy as np
2
3   # input data
4   x=np.array([ [0,0,1],
5                [0,1,1],
6                [1,0,1],
7                [1,1,1] ])
8
9   # output labels
10  y=np.array ([[0,0,1,1]]).T
11
12  # initialize weights randomly with mean 0
13  weights = 2*np.random.random((3,1))-1
14
15  #update weights (training/backpropagation)
16  for iter in xrange(1000):
```

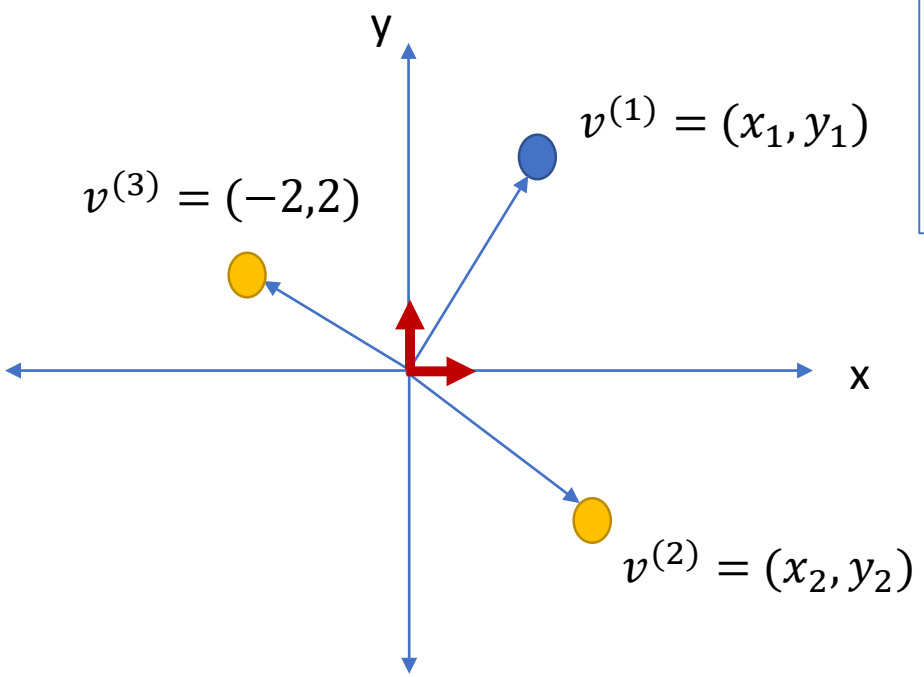
Data

Weight Vector

```
15  #update weights (training/backpropagation)
16  for iter in xrange(1000):
17
18   # forward propagation
19   l0=x
20   l1=nonlin(np.dot(l0,weights))
21
22   # how much did we miss?
23   l1_error=y-l1
24   # multiply how much we missed by the
25   #slope of the sigmoid at the values in l1
26   l1_delta=l1_error*nonlin(l1,True)
27
28   #update weights
29   weights+=np.dot(l0.T,l1_delta)
```

New Vector

# Basis Vectors and Linear Combinations




Let $\hat{x} = \begin{pmatrix}1\\0\end{pmatrix}$ and $\hat{y} = \begin{pmatrix}0\\1\end{pmatrix}$ be a unit vector (of magnitude 1) aligned with the x and y axis

Terminology Alert #1: Such unit vectors aligned with the axes i.e. $\hat{x}$ and $\hat{y}$ are called "BASIS VECTORS"

Question: Now $v^{(1)}$ can be represented in terms of $\hat{x}$ and $\hat{y}$ → How?

$Ans: v^{(1)} = (x_1 . \hat{x}) + (y_1 . \hat{y})$

$v^{(3)} = (-2 . \hat{x}) + (2 . \hat{y})$

Terminology Alert #2: The above representation is called a LINEAR COMBINATION

Conclusion:
Any vector can be represented as a linear combination of its basis vectors

# Linear Combination in Data Science

Any training data instance can be represented as a linear combination

Example the instance $v^{(1)} = (x_1 . \hat{x}) + (y_1 . \hat{y})$

Mathematical Meaning: Linear combination is obtained by stretching the $\hat{x}$ and $\hat{y}$ basis vectors with scalar values $x_1$ and $y_1$ (sum of two scaled unit vectors)

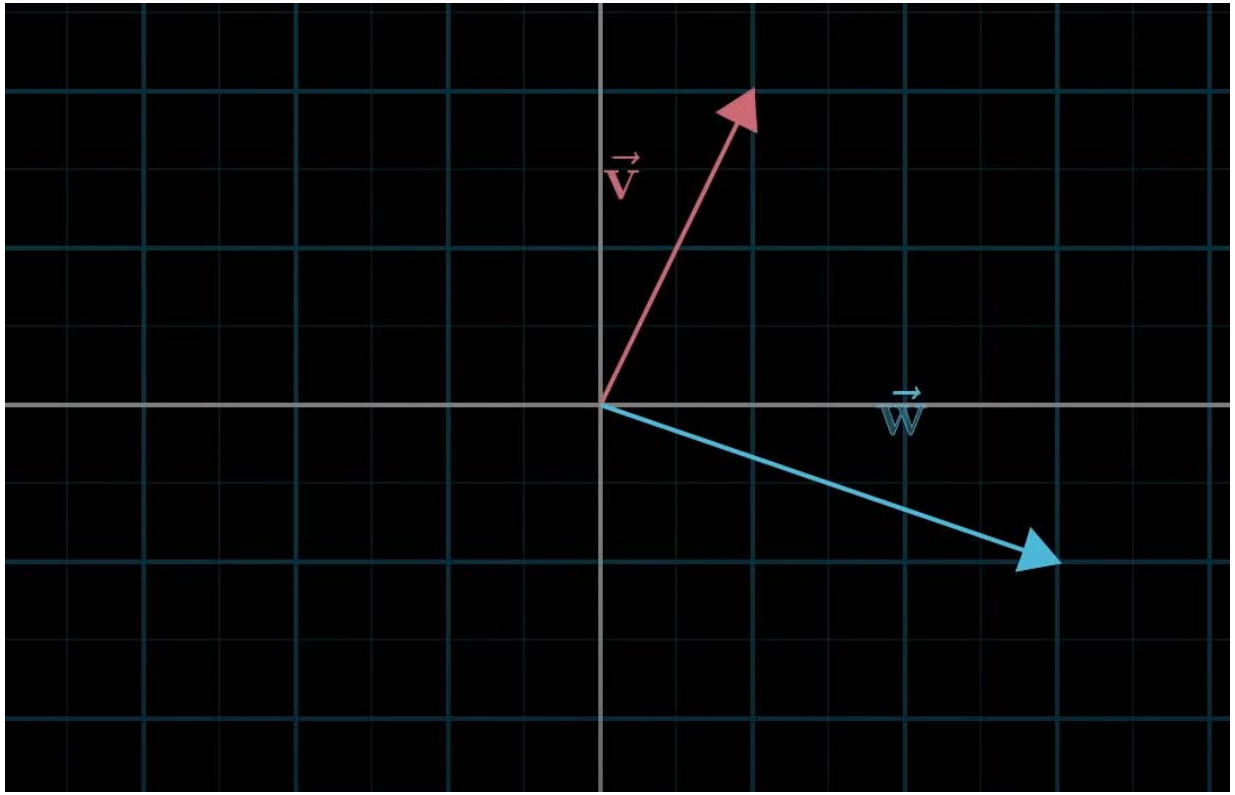
Physical meaning: $v^{(1)}$ is composed of $x_1$ parts of feature $\hat{x}$ and $y_1$ parts of feature $\hat{y}$

Two alternatives to visualize multiple training data instances (the training set)

(i) Points in the feature space (the axes)

(ii) A list of vectors

# Basis Vectors "Choice" could be arbitrary




Instead of unit vectors aligned with the axes,

We could have picked virtually any set of vector as our basis vectors (e.g., $\hat{v}$ and $\hat{w}$)

and represent all other points in the dataset as a linear combination of these two new basis vectors $\hat{v}$ and $\hat{w}$

It will still work the same

Note $\hat{v}$ and $\hat{w}$ not aligned with axes of the original coordinate system

Terminology alert !

In $R^2$ $\hat{x} = \begin{pmatrix}1\\0\end{pmatrix}$ and $\hat{y} = \begin{pmatrix}0\\1\end{pmatrix}$ are called "standard basis" (which are also orthonormal i.e., perpendicular to each other)

$\hat{v}$ and $\hat{w}$ are orthonormal wrt to each other but Not wrt to the standard basis

Conclusion:
Basis vectors are a matter of choice. One can take liberties according to the nature of the problem

# Understanding the Span

- Definition:

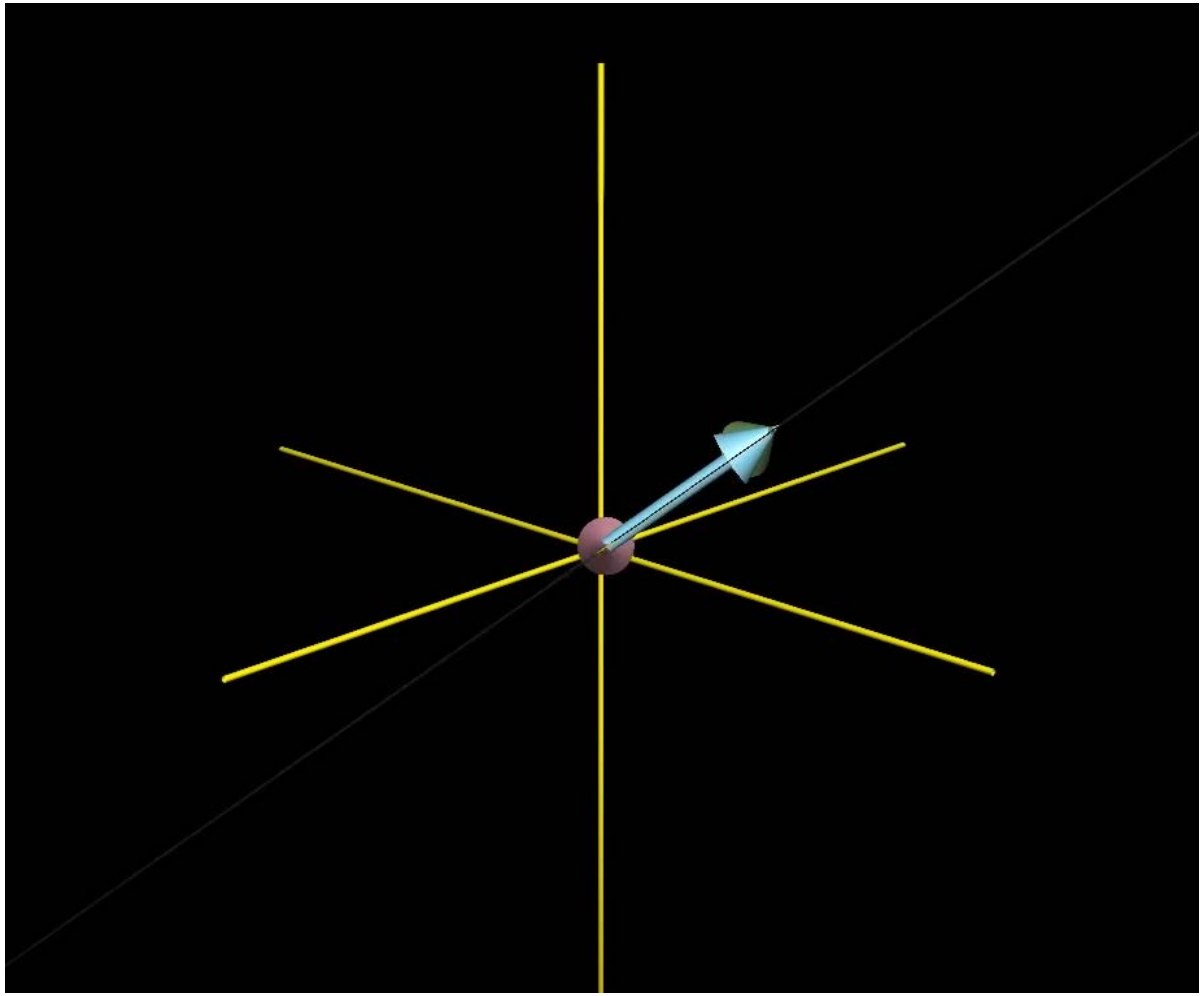
  set of all linear combinations (nothing but points or vectors)

  that you can potentially reach given a set of vectors

- Meaning:

  Given any set of vectors (say two), what is the set of points can you reach in this coordinate system? In $R^2$ if no constraints are given, the two standard basis vectors will produce a span equivalent to a 2D plane sheet which is infinite. In reality though, often there are constraints.

# Illustration of Span



Question: in $R^3$ the standard basis $\hat{x} = \begin{pmatrix}1\\0\end{pmatrix}$ and $\hat{y} = \begin{pmatrix}0\\1\end{pmatrix}$ will give a span equal to ?

*Ans: A plane sheet cutting through the origin.*

# Why we need span in Data Science

- Given a set of vectors, what can you do with them?
- You can add (subtract) or multiply by a scalar

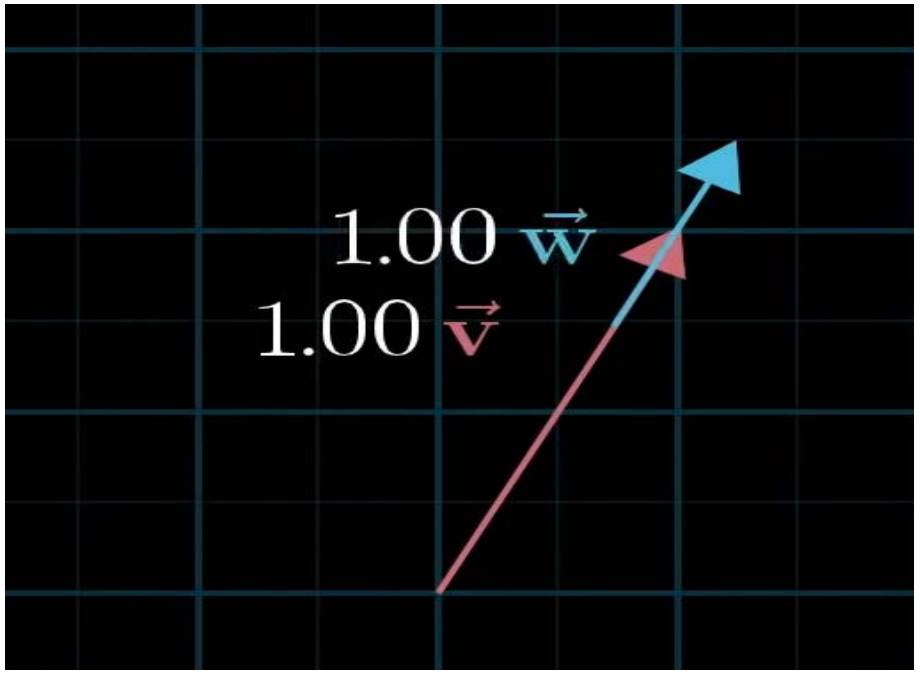
- You're given a list of vectors, and told you can only play with these vectors.
- See all possibilities you can make with them. The set of all things you can make *is the span of those given vectors.*

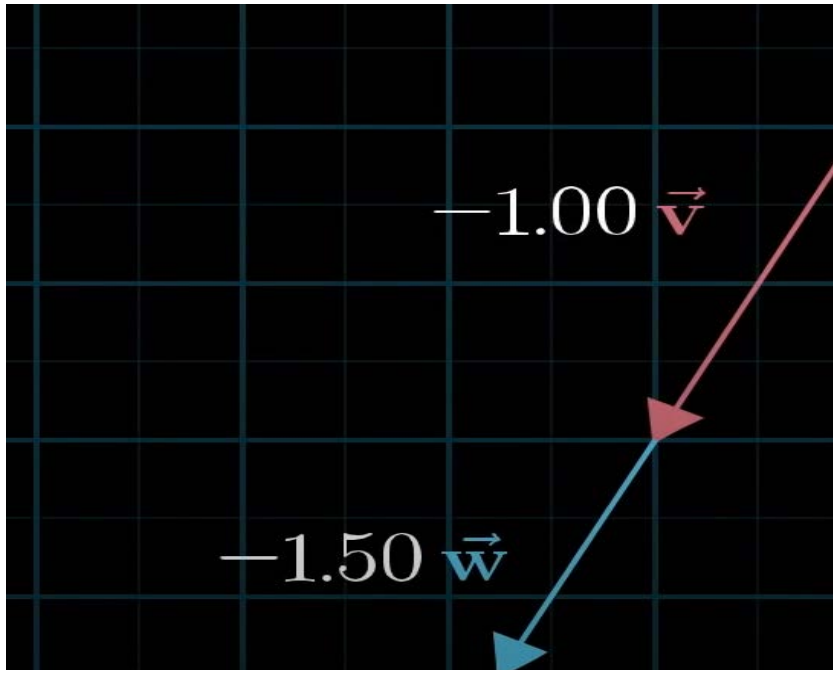
- That the span is a subspace (subset) is nice → reduces search space for one

- it's always good to have objects that are *closed* under certain operations, and subspaces are just that: closed under vector addition and scalar multiplication.

- This isn't true for most generic sets of vectors, but definitely true for the span of a set of vectors. So spans have nice properties.

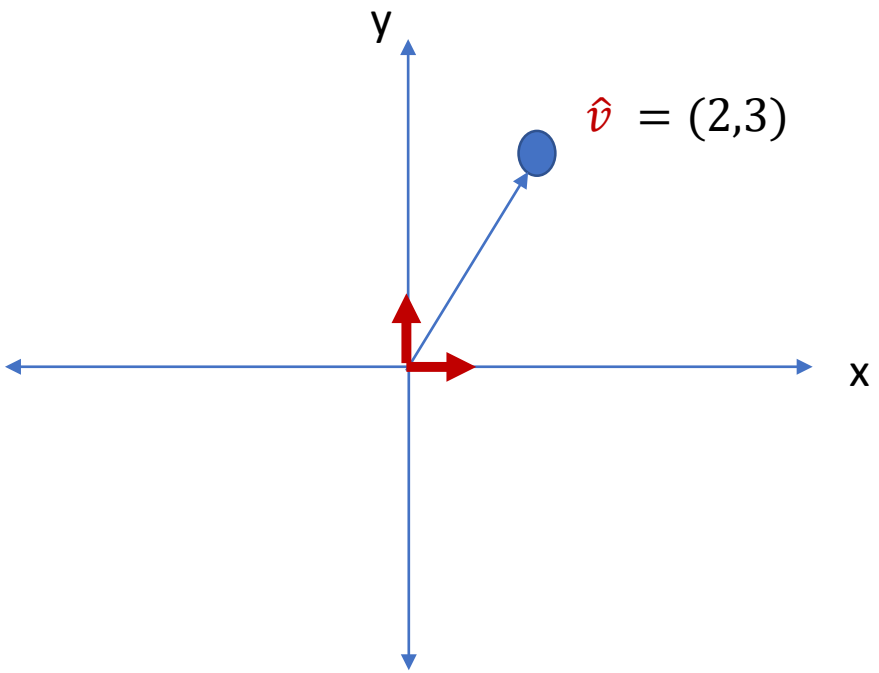
# Special case when given vectors line up




Note if you are given two vectors that
Line up, the set of all linear combinations
Addition scalar multiple, now will give you
SPAN = Just a line that is aligned with these two vectors only
not a 2D plane sheet

# Reinforcing Linear Dependence and Span




Suppose a third vector $\hat{v}$ is on the span of your previous two vectors $\hat{x}$ and $\hat{y}$

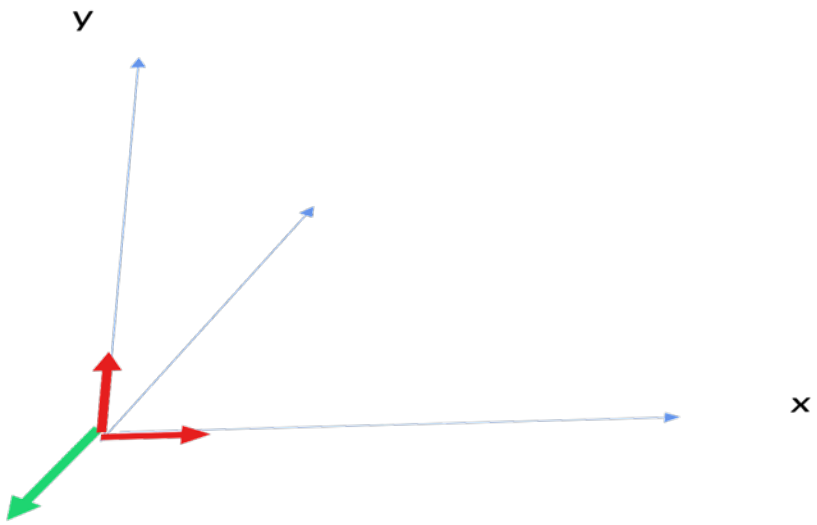
How? → 2 $\hat{x}$ + 3 $\hat{y}$ = $\hat{v}$

$\hat{v}$ does not add to the span
(no new points can be reached )

Terminology Alert !
All such vectors $\hat{v}$ are called linearly dependent on the previous two vectors $\hat{x}$ and $\hat{y}$

Linearly dependent vectors
(i) Do not add to the span
(ii) Can be expressed as a linear combination of other vectors

# Reinforcing Linear Independence and Span




$\hat{v}$ protruding up from the xy plane
Unlocks another (*third*) dimension

Suppose a third vector $\hat{v}$ is not on the span of your previous two vectors $\hat{x}$ and $\hat{y}$

$\hat{v}$ adds to the span
(a whole new points can be reached using $\hat{x}$, $\hat{y}$and $\hat{v}$ )

Terminology Alert !
All such vectors $\hat{v}$ are called linearly independent of the previous two vectors $\hat{x}$ and $\hat{y}$

Linearly independent vectors
(i) Adds to the span
(ii) Cannot be expressed as a linear combination of other vectors
(iii) Basis vectors need to be linearly independent to span the whole vector space

# SPAN and Linear Dependence in Data Science

If a vector is redundant and can be expressed as a combination of the first two; i.e. linearly dependent →

I can ignore use of new variables while doing analysis

A form of reduction while making sense of big data with lot of points

If a vector is not the span and it expands the span of the previous two vectors (adds a dimension), this kind of third vector is called as linearly independent w.r.t the previous two vectors (because I cannot ignore this third vector)

# Linear Transformations

- Linear Transformation is essentially a function in linear algebra
- They take in a input vector x and produce an output vector y
- x → (linear transformation) → y
- Geometry: Input vector moves over to its corresponding output → a notion of bending the vector space

# Transformation Contd..

- In linear algebra, transformation of the vector space is linear
- Meaning

    1. origin remains the same before and after transformation

    2. the grid lines of the vector space are parallel and evenly spaced across either side of the transformation

# Matrices

- Matrix → a way of packing information
- i.e. taking in a vector $\begin{pmatrix} x=5 \\ y=7 \end{pmatrix}$
- I want to get an output or have an output vector $\begin{pmatrix} a \\ b \end{pmatrix}$
- Find a matrix such that x. m11 + y. m21 = $\begin{pmatrix} a \\ b \end{pmatrix}$

  x. m12 + y. m22 =

- 5 $\begin{pmatrix} m_{11} \\ m_{21} \end{pmatrix}$ + 7 $\begin{pmatrix} m_{12} \\ m_{22} \end{pmatrix}$ = 5 m11 + 7. m21

  5. m12 + 7. m22

$$\mathrm{M} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

# Matrices for Transformation

- First column of the matrix M → where the first basis vector will land after transformation

- Second column of the matrix M → where the second basis vector will land after transformation

- Interpretation1
- Matrices can be transformation of basis vectors

# Matrices

- Apart from interpreting matrices as linear transformations there is another very important aspect

- Matrices are a compact way for storing data containing multiple features (the columns) and huge number of training examples (rows)

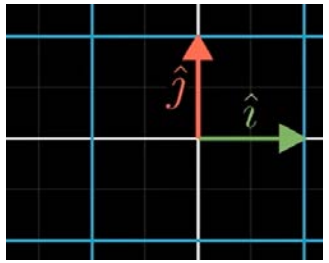
# Determinant of Matrix

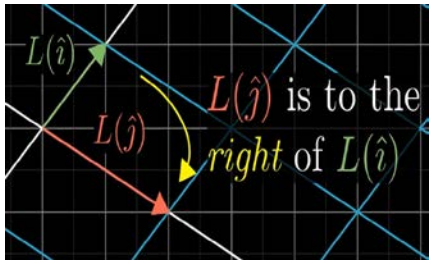
Determinant of a matrix A (det A)

quantifies the factor by which the area changes (increases or decreases)
by a linear transformation specified by a matrix A

Det A = 0 → if the transformation squishes the vectors onto a line or a point (in 2D) or a region with no volume

Det A is negative if the space/orientation is flipped




Det A is positive if the transformation cannot squish vectors onto a line or a point or a lower dimension compared to the input space

# Understanding Det(A)=0

Since a matrix is a transformation, it causes an input vector to land on some output vector.

Now if the determinant of the matrix (transformation) is zero, it means that the area/volume of the transformed output vector space, is not there.

When determinant of a matrix is zero, it means that the output vector is a line, point, or a plane

# Inverse of Matrices

Say $\hat{x}$ is a vector of variables

A corresponds to some linear transformation that bends space

$$A^{-1}Ax = A^{-1}v$$

we are looking for a vector $\hat{x}$ (nothing but a point) which after transformation by matrix A lands on a pre-specified vector $\hat{v}$

$$x = A^{-1}v \qquad A^{-1}A = I$$

Playing a transformation in reverse with $\hat{v}$ to see where it lands; wherever it lands is $\hat{x}$

When $\det(A) = 0$, there is a no inverse

# Some Intepretations of Matrix Transformations

$Ax = v$

Suppose you apply inputs $\hat{x}$ in a system and observe an output $\hat{v}$ . The inherent nature of a system transforms the $\hat{x}$ to $\hat{v}$ . In such case, we can solve for A from the Ax = v ; A may give how much each input feature contributes to the observed output; a transfer function

Suppose you know the output $\hat{v}$ in a system and know how the system behaves specified by A. However, there are some uncertainties. Playing transformation in reverse with $A^{-1}$ and $\hat{v}$ , one can get an approximate idea of the values of the features next time

# Rank of a matrix

- Solutions are harder to exist when the transformation squishes points onto a lower dimension
- This interesting aspect has some fancy terminology known as RANK
- When output via a matrix transformation is a line (i.e. one dimensional), we say that this matrix transformation has a RANK = 1
- Similarly, if the output via a matrix transformation is a 2D plane, then its RANK = 2 and so on;
- In general, the term RANK says → the number of dimensions in the "output" found from a matrix transformation

# Rank and Column Spaces

- Set of all possible outputs for a matrix transformation is known as → column space
- Remember that columns of a matrix (transformation) say → where your basis vectors land after this transformation is used
- Span of transformed vectors above gives all possible outputs
- Column space is the span of the columns of your matrix

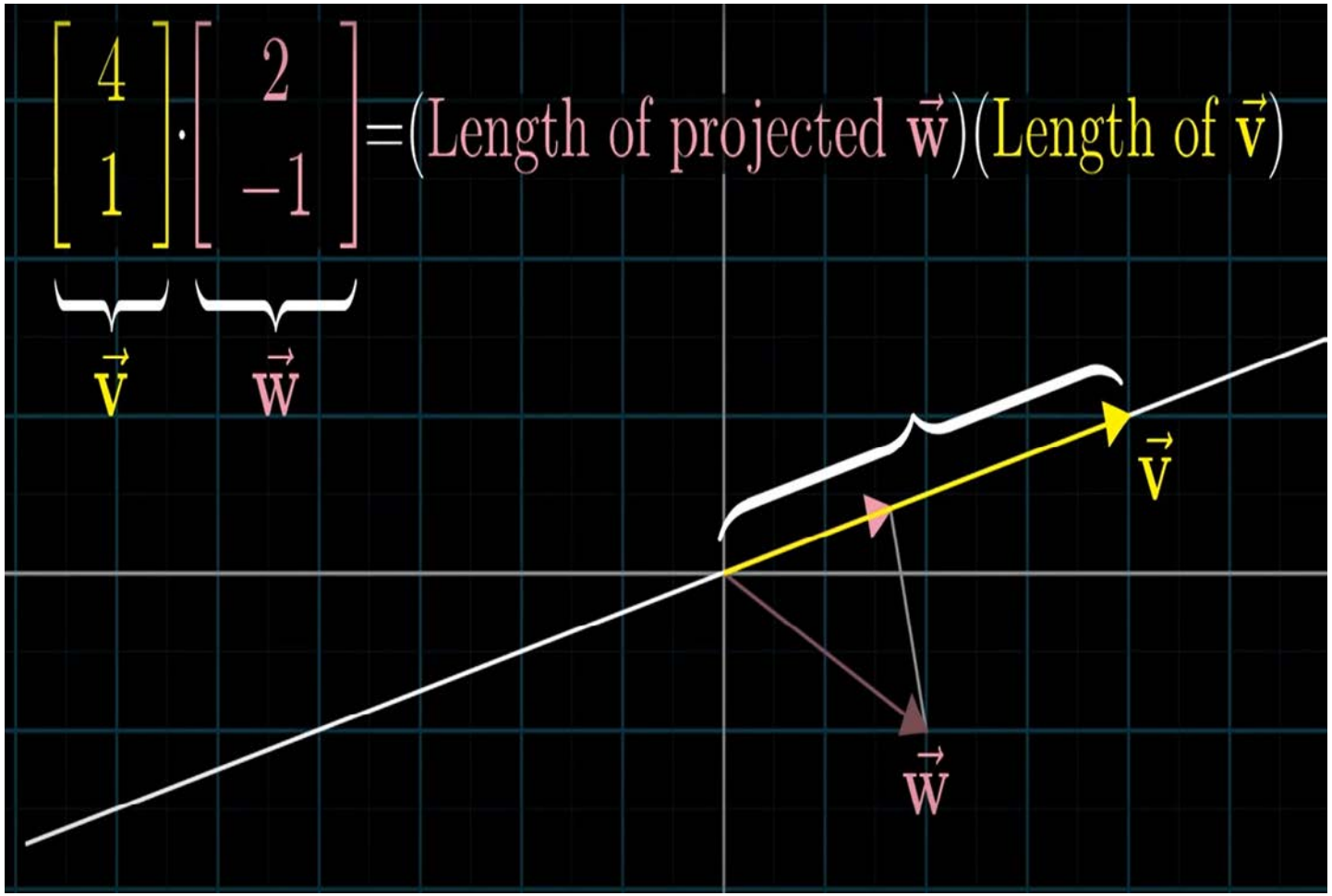
# Null Spaces

- Set of vectors that land on the origin (zero vectors) → Null space

$$Ax = v \Rightarrow Ax = 0$$

When v happens to be a 0 vector $\binom{0}{0}$ , the null space gives you all possible solutions of the equation.

# Dot products




Dot product of two vectors ($\hat{v}.\hat{w}$) =

length of "projection" of the 2nd vector ($\hat{w}$) onto the first vector $\hat{v}$

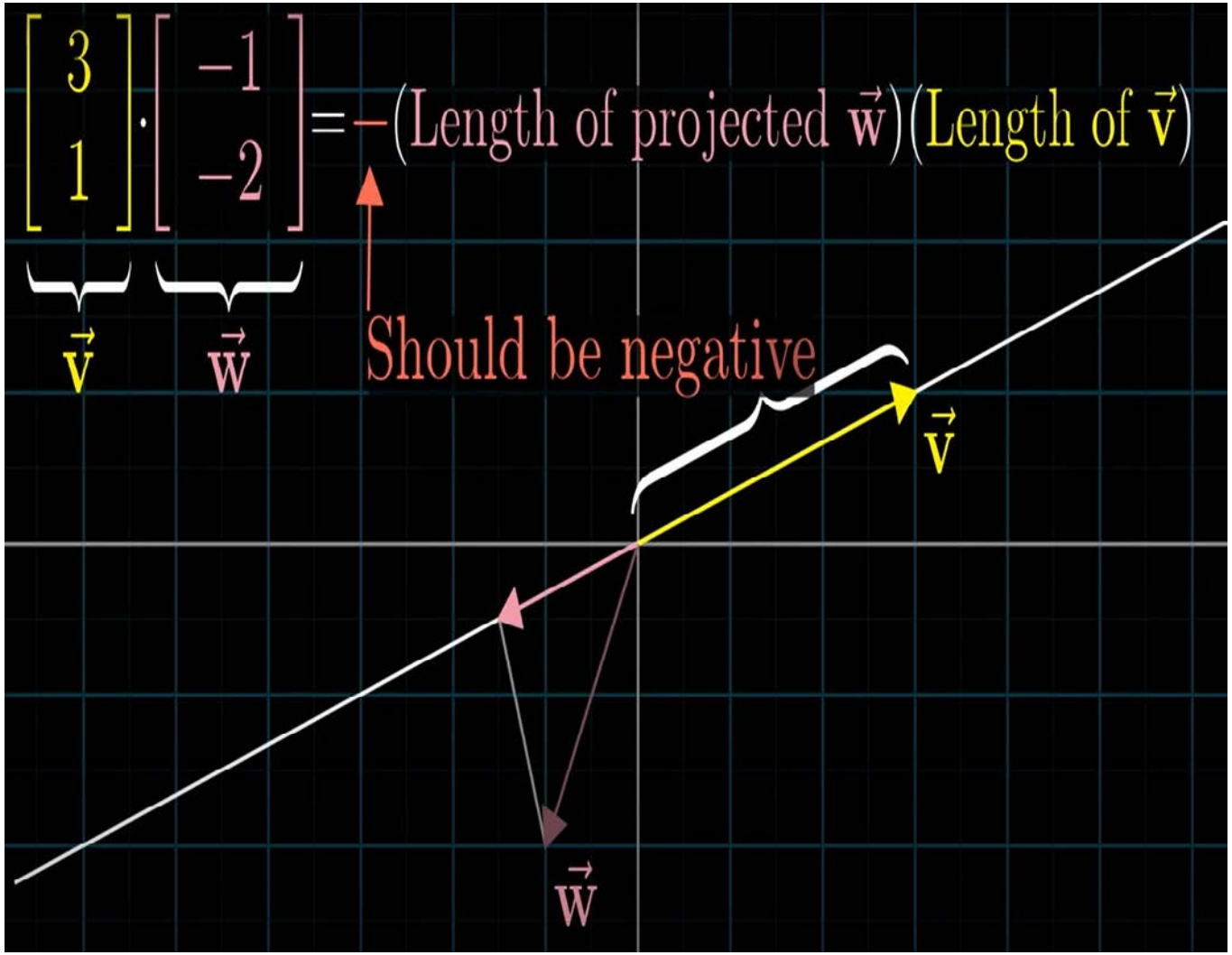
x

length of the first vector (v)

Note: order does not matter on which vector is projected

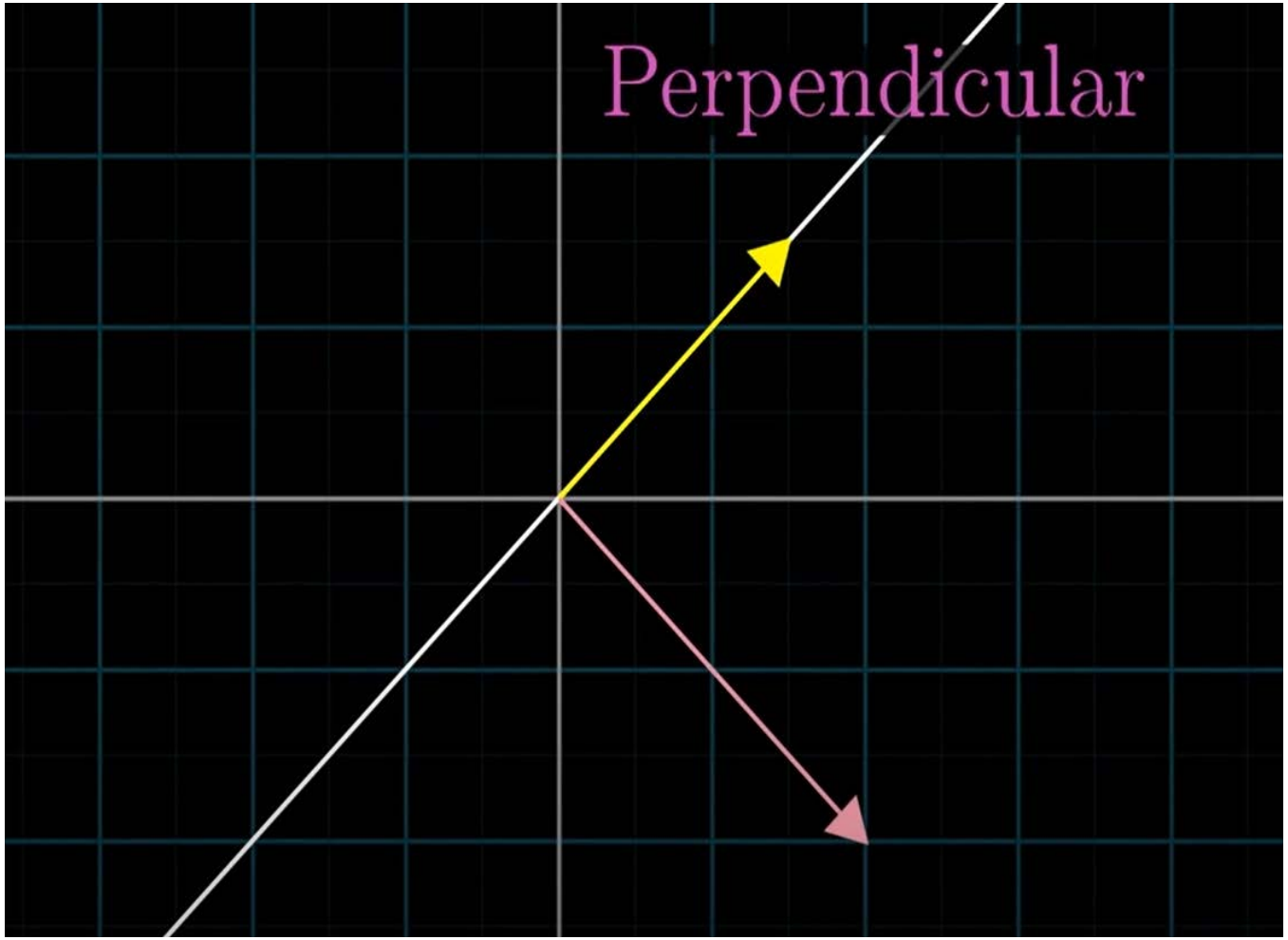
When two vectors are generally pointing in the same direction, their dot product is positive

# Dot products contd..




When two vectors are generally pointing in the opposite direction, their dot product is negative

# Dot products




When two vectors are perpendicular

(Note this can be viewed as neither same nor opposite direction)

The dot product is zero

# What can dot products tell → Indicate correlation

# DUALITY of DOT products and Matrix vector Multiplication

- Linear transformations → those which take in vectors in multiple dimensions (say from 2D or above) and produce an output to 1D (a single number on the real number line  i.e. from vectors to numbers)

- This is similar to multiplying a 1x 2 matrix and a 2x 1 matrix which gives a single number (much like matrix vector multiplication)

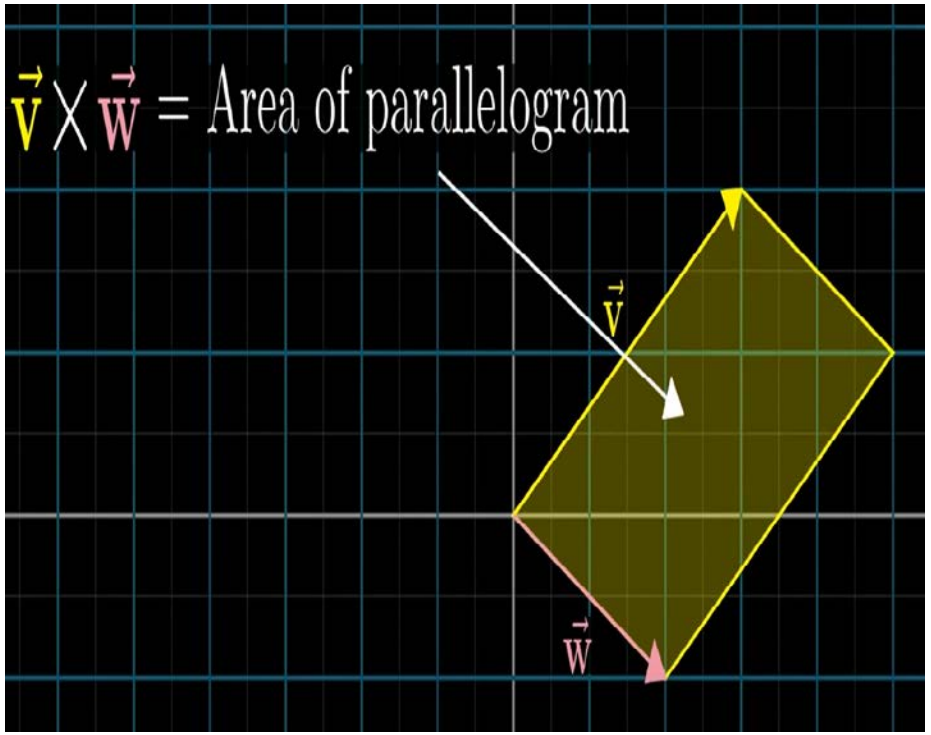
- 1x 2 matrices are analogical to 2D vectors → DUALITY

- Dot product is similar to matrix vector multiplication

# Duality contd..

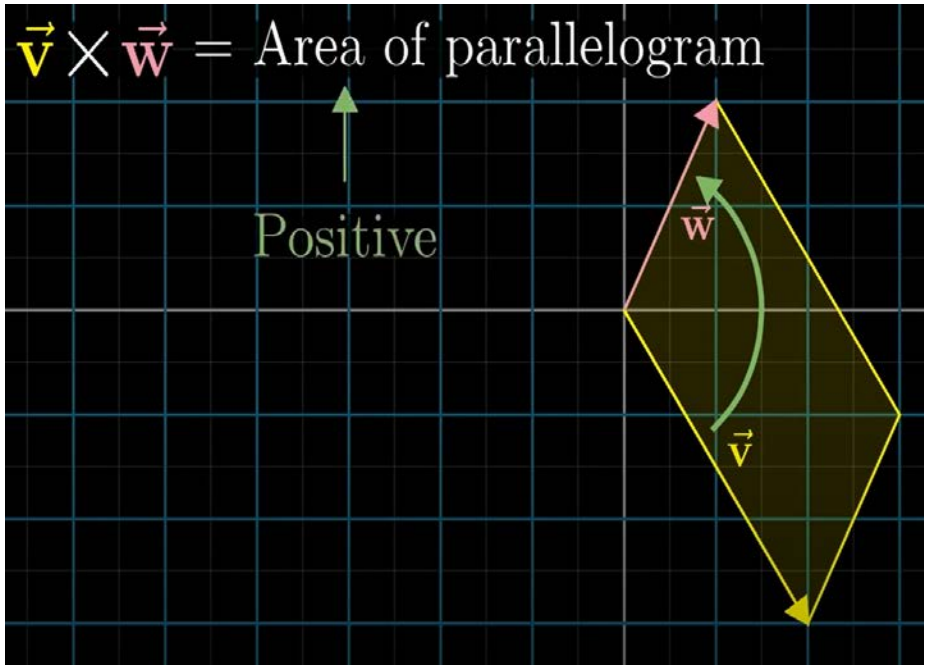
- Dual of a vector → the linear transformation that it encodes
- Dual of a (matrix) linear transformation in a one d space→ is a certain vector in that one d space
- So vectors can be viewed as an embodiment of a *linear* transformation, and not merely a single data point in a coordinate system

# CROSS PRODUCTS

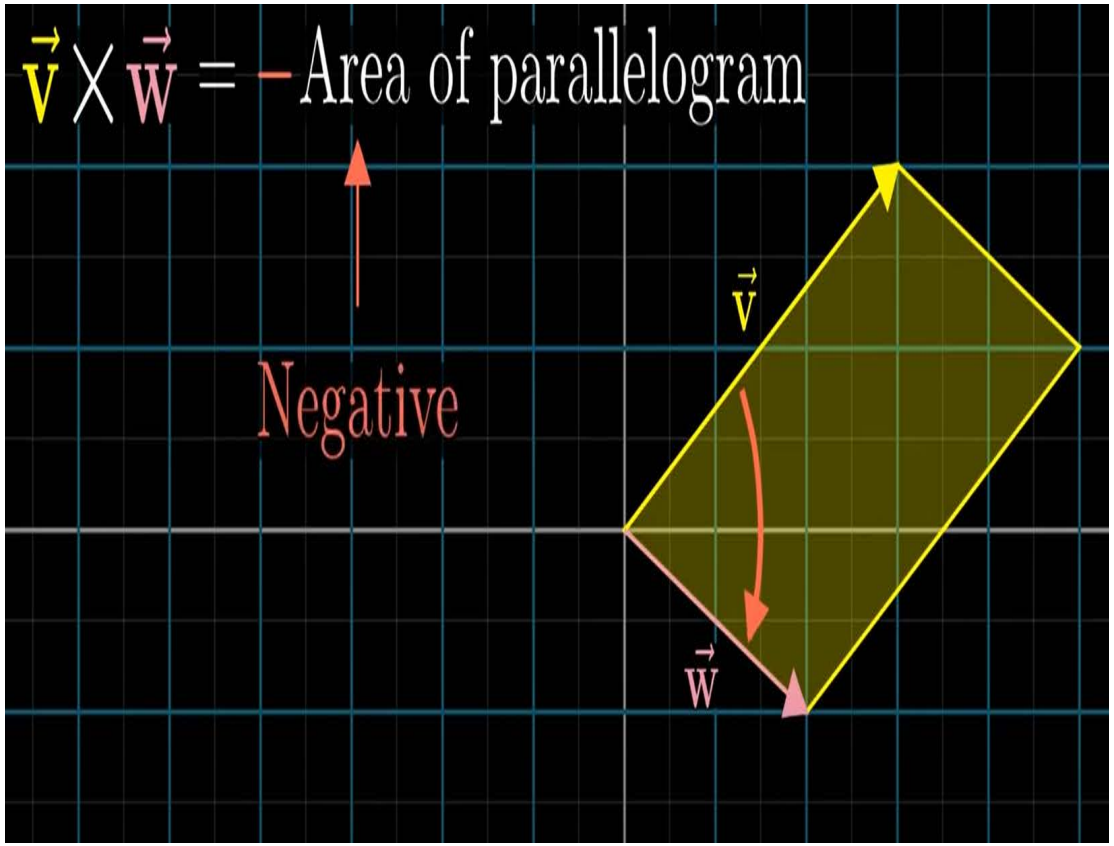



Unlike dot products in cross products "order Matters"→ $\hat{v} \quad X \quad \hat{w} = -\hat{w}$ X $\hat{v}$

V is on right of W (counter-clock rotation) → area is positive
Negative otherwise



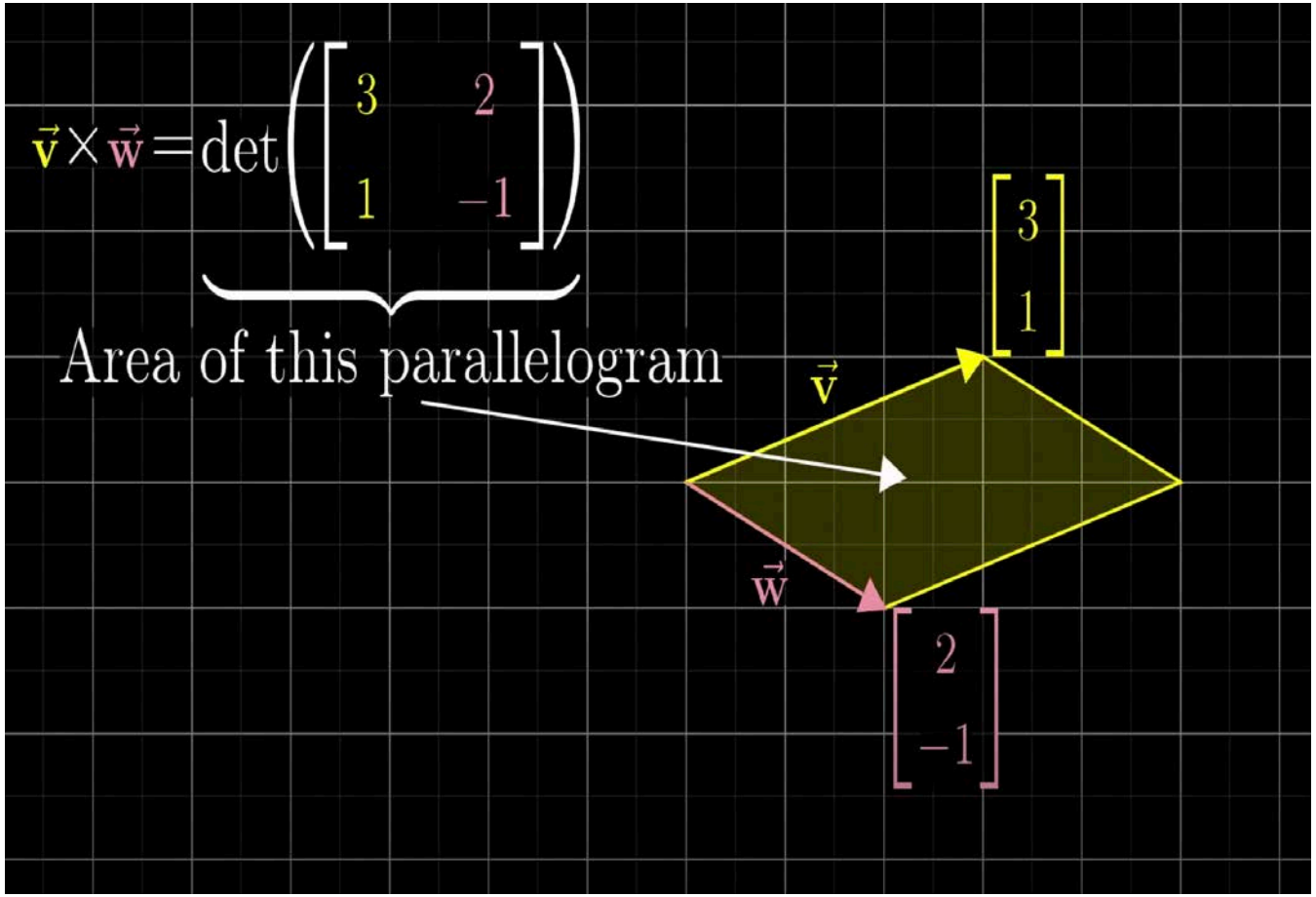
# CROSS PRODUCTS




If V is on left of W (counter-clock rotation) →
area is negative

So, $\hat{v}\ X\ \hat{w} = -\ \hat{w}$ X $\hat{v}$

# Compute Cross Product




For 2D cross product $\hat{v}\ X\ \hat{w}$, we write the coordinates of $\hat{v}$ and $\hat{w}$ as the first and second column of the matrix respectively. Then we just compute the determinant.

NOTE: Here the determinant represents the factor by which the area of this parallelogram is changed.

# Compute  Cross Product contd..

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \times \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \det\left(\begin{bmatrix} \hat{i} & v_1 & w_1 \\ \hat{j} & v_2 & w_2 \\ \hat{k} & v_3 & w_3 \end{bmatrix}\right)$$

For a 3D cross product between $\hat{v}\ X\ \hat{w}$, the second and third columns of the matrix contain the coordinates of $\hat{v}$ and $\hat{w}$ respectively and the first column contains the basis vectors. Then we just compute the determinant.

# Cramer's Rule

- A convenient method to solve a linear system of equations for just one single variable without having to solve the whole system of equations.

- Let's consider the following system of equations:

$$a_1 x + b_1 y = c_1$$
$$a_2 x + b_2 y = c_2$$

Let *D* be the determinant of the coefficient matrix and $D_x$ be the determinant formed by replacing the x column with the constant column.

- Using Cramer's rule:

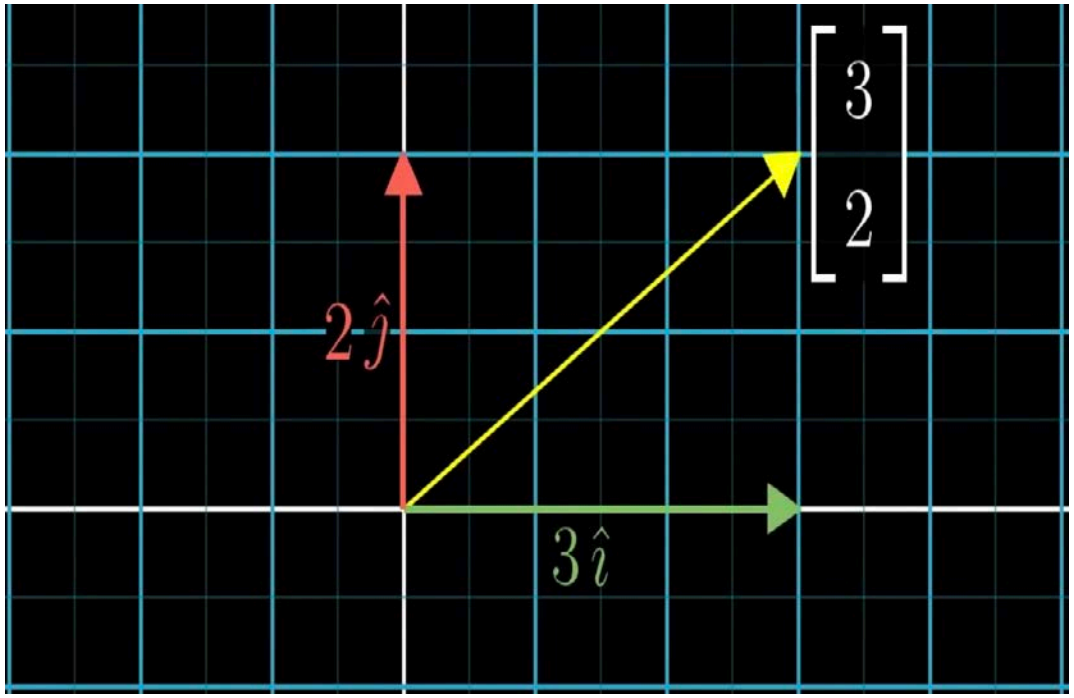
$$x = \frac{D_x}{D} = \frac{\begin{bmatrix} c_1 & b_1 \\ c_2 & b_2 \end{bmatrix}}{\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}}, \; D \neq 0$$

# Cramer's Rule contd..

- Similarly, while solving for y, the y column is replaced with the constant column.

$$y = \frac{D_y}{D} = \frac{\begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix}}{\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix}}, D \neq 0$$

# Change of Basis

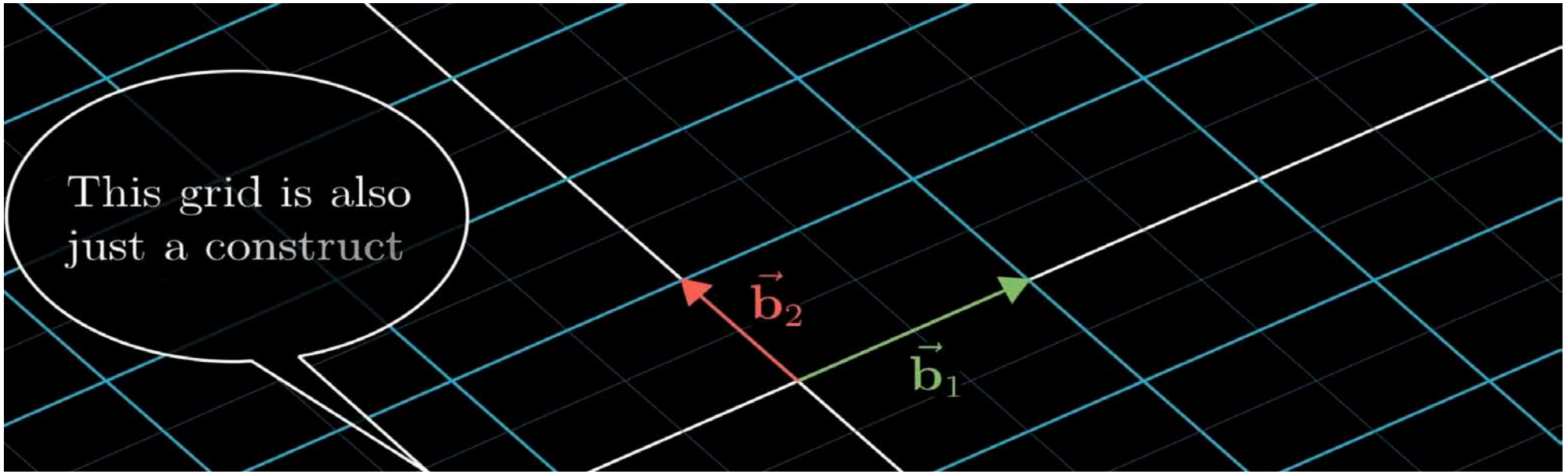



- A vector sitting in a 2D space can be described with coordinates. We can think each of the numbers as a scalar that stretches or squishes vectors.
- If $\hat{\imath}$ and $\hat{\jmath}$ are basis vectors, the first coordinate scales $\hat{\imath}$ and the second coordinate scales $\hat{\jmath}$.

**Question:** What if we used different basis vectors in a different grid ??

# Change of Basis contd..

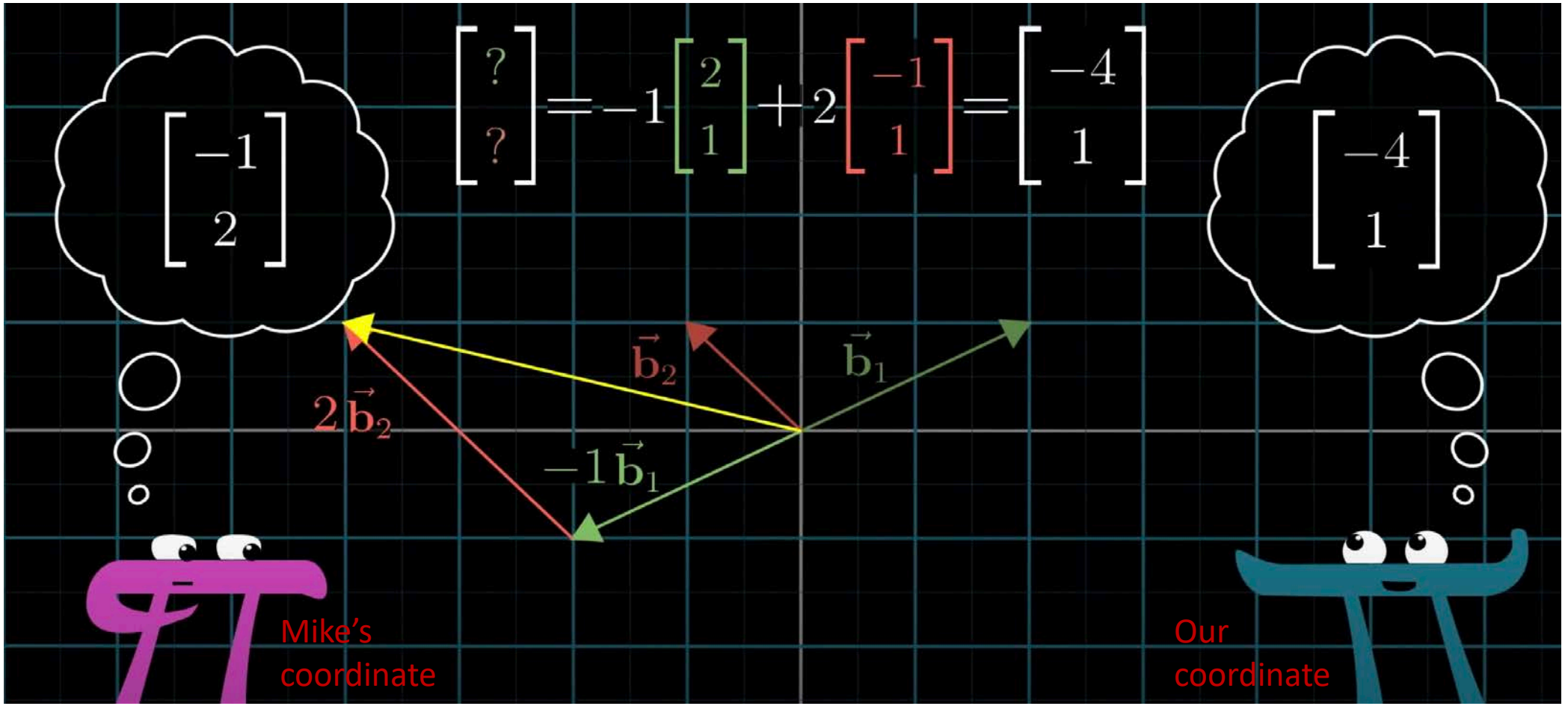
- Space does not have a particular system of grid. So, someone might draw their own grid in the space with a fixed origin.




- A vector in one grid(coordinate system) is different in another grid (coordinate system) depending on the choice of the basis vectors.
- Now the question is: **How do we translate between coordinate systems?**

# Change of Basis contd..

- Let’s say Mike has a different coordinate system than ours. In order to translate a vector from Mike’s coordinate system to our coordinate system, we have to scale each of his basis vectors by the corresponding coordinates of the vector of our system and add them together.

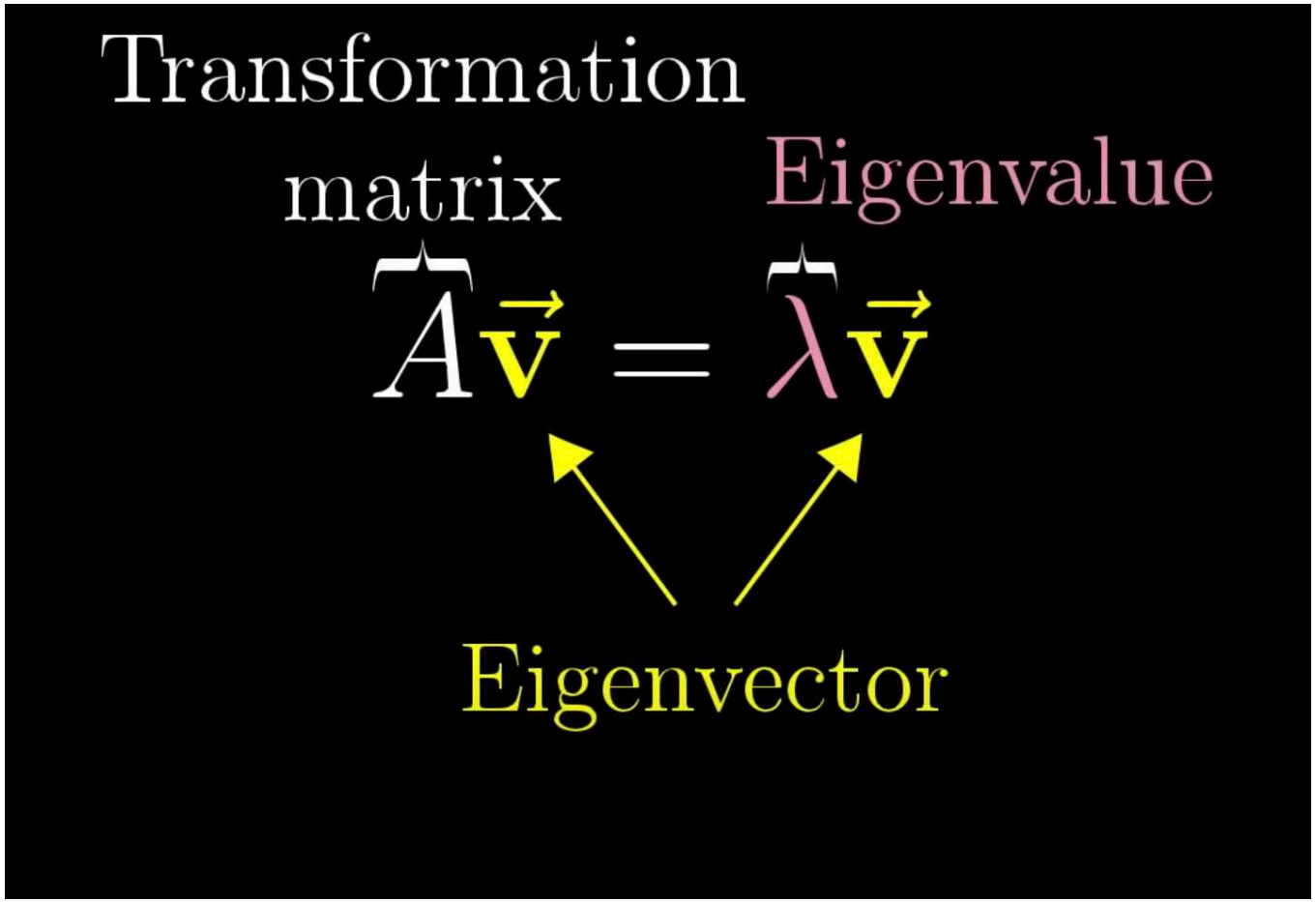


# Eigenvector and Eigenvalue

- Let **A** be a square matrix. Then a nonzero vector $\vec{v}$ is an eigenvector of **A** if there exists a scalar $\lambda$ such that

$$\mathbf{A}\,\vec{v} = \lambda\,\vec{v}$$

- The scalar $\lambda$ is known as the eigenvalue corresponding to the eigenvector.



# Eigenvector and Eigenvalue contd..

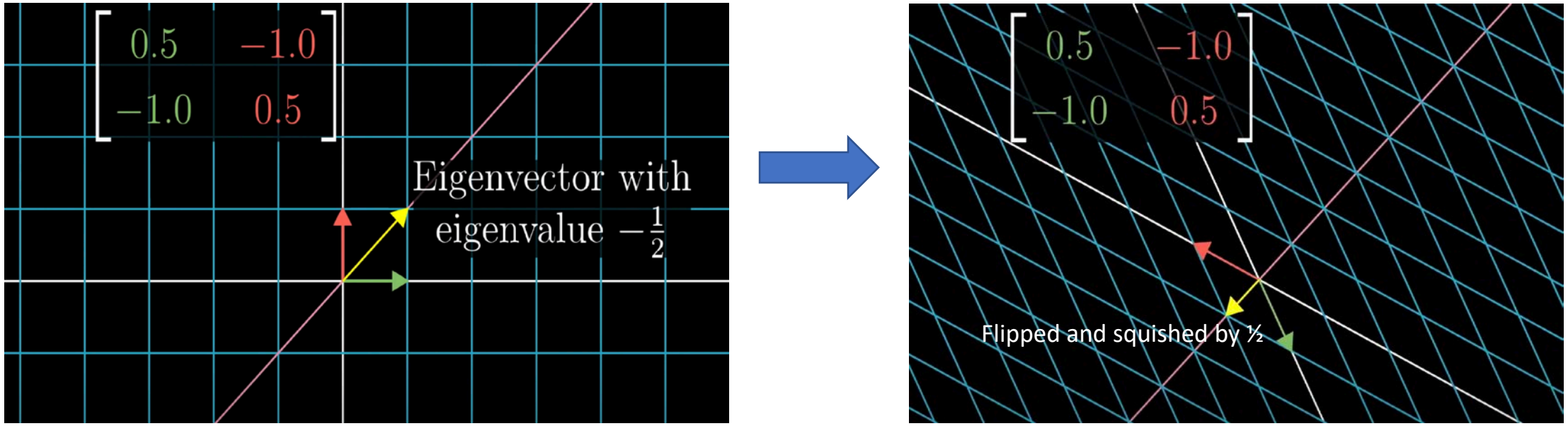
- During transformations, eigenvectors remain in their own span.
- A matrix can only stretch or squish these vectors like a scalar.
- The factor by which an eigenvectors gets stretched or squished is called its corresponding eigenvalue.

# Eigenvector and Eigenvalue contd..

- **Question:** Can eigenvalues be negative?

☞ Yes, eigenvalues can be negative. An eigenvector with an eigenvalue of $^{-1}/_{2}$ (the yellow vector)means that the vector gets flipped and squished by a factor of $^{1}/_{2}$ .

**NOTE:** Although the vector gets flipped and squished by a factor of $^{1}/_{2}$ , it stays on the same line in its span without getting rotated off of it.



# Eigendecomposition

- When we break any mathematical objects into their constituent parts or find their properties, we can understand them better. For example, we can understand the true nature of an integer when we decompose it into prime factors.
- Similarly, when we decompose matrices, we can learn about their functional properties which is not evident when we represent them as an array of elements.
- One of the widely used matrix decompositions is eigendecomposition in which we decompose a matrix into a set of eigenvectors and eigenvalues.

# Eigendecomposition contd..

- Suppose that a square matrix $\boldsymbol{A}$ has n linearly independent eigenvectors, $\{v^1, \ldots, v^n\}$, with corresponding eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$.

- We may concatenate all of the eigenvectors to form a matrix $\boldsymbol{V}$ with one eigenvector per column: $\boldsymbol{V} = [v^1, \ldots, v^n]$.

- Similarly, we can concatenate the eigenvalues to form a vector $\lambda = [\lambda_1, \ldots, \lambda_n]^\top$.

- If $\Lambda$ is a diagonal matrix, then the eigendecomposition of A is given by:

$$\boldsymbol{A} = \boldsymbol{V} \Lambda \boldsymbol{V}^{-1}$$

# Eigendecomposition contd..

- **What does Eigendecomposition tell us about a matrix?**
- A matrix is singular if and only if any of the eigenvalues are zero.
- If eigenvalues are all positive, then the matrix is called positive definite.
- If eigenvalues are all positive or zero-valued, then the matrix is called positive semidefinite.
- If eigenvalues are all negative, then the matrix is called negative definite.
- If eigenvalues are all negative or zero-valued, then the matrix is called negative semidefinite.

- [source: Deep Learning]

# Singular Value Decomposition

- Eigendecomposition works only if a matrix is square. So when a matrix is not square we use singular value decomposition.
- Singular value decomposition is a commonly used method for decomposing a matrix into three other matrices.
- In other words, the singular value decomposition is the factorization of an $n \times m$ matrix A as the product $A = U\Sigma V^{\top}$ where U and V are orthogonal matrices and $\Sigma$ is a diagonal matrix(NOT necessarily a square matrix).
- The diagonal entries, $\sigma_1 \geq \sigma_2 \geq \ldots\ldots \sigma_m \geq 0$, are called singular values of A. The columns of U are called the left-singular vectors and the columns of V are called the right-singular vectors of A.

# Singular Value Decomposition contd..

- We can actually interpret the singular value decomposition of A in terms of the eigendecomposition of functions of A. The left-singular vectors of A are the eigenvectors of $AA^\top$. The right-singular vectors of A are the eigenvectors of $A^\top$A. The non-zero singular values of A are the square roots of the eigenvalues of $A^\top$A. The same is true for $AA^\top$.

- One of the most useful features of the singular value decomposition is that we can use it to partially generalize matrix inversion to non-square matrices.

[source: Deep Learning]

Helpful Resources and References

This document uses some snapshot geometrical pictures from
youtube channel of 3blue1brown for geometry of linear algebra
https://www.youtube.com/channel/UCYO_jab_esuFRV4b17AJtAw
Please check it out for other geometric interpretations beyond AI and data science

Check out Linear algebra materials by Prof. Zico Kolter for mathematical formulaes and proofs
https://www.cs.cmu.edu/~zkolter/course/15-884/linalg-review.pdf