CPE 490 590 — Linear Algebra Review

Linear Combination

N features.

$w_i x_i \rightarrow$ Linear Combination

w_i = weight of each f

x_i = feature

System of Linear Equations:

$$
\begin{aligned}
a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n &= b_1 \\
a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n &= b_2 \\
&\vdots \\
a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n &= b_m
\end{aligned}
$$

Vector

Coloumn vector $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \rightarrow$ 3 rows 1 column

Row vector $\vec{v} = [v_1, v_2, v_3]$

```
>> import numpy as np
>> np.array([1,2,3]) # numpy array.
>> a = [1,2,3]        # list
```

Scalar field

A function that assigns a number to each point in a region or a space or space time.

E.g. Temperature distribution

$$f(x,y,z) = x^2 + y^2 + z^2$$

$\mathbb{K} \rightarrow$ Scalar field

Vector Space

A set V is called a vector space over $\mathbb{K}$ if it satisfies

1. For any vector $\vec{x}$ and $\vec{y}$ we have $\vec{x} + \vec{y} \in V$

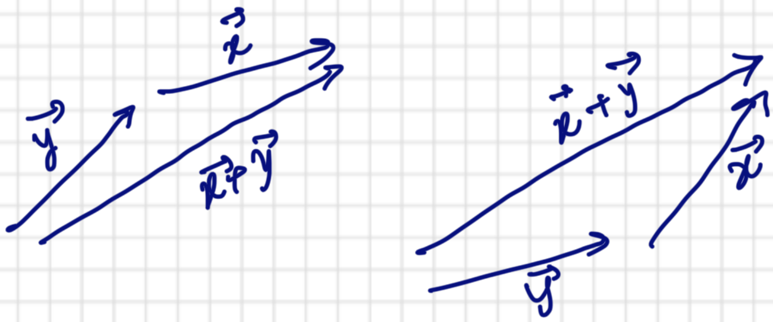
2. For any vector $\vec{x}$ and a scalar a $a\vec{x} \in V$

3. There is an existence of $\vec{0} \rightarrow$ zero vector

4. For any $\vec{x}$, we have $-\vec{x}$.

$\downarrow$

Inverse vector.



Vector Sum

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

Subspaces

A subset W of V is a subspace if W itself is a vector space over $\mathbb{K}$ with vector sum, scalar multiple, zero vector and inverse vector defined in V.

i.e.

(i) $\vec{x}+\vec{y} \in W$ for $\vec{x} \in W$, $\vec{y} \in W$

(ii) $a\vec{x} \in W$ for $a \in \mathbb{K}$, $\vec{x} \in W$.

(iii) $\vec{0} \in W$

(iv) $-\vec{x} \in W$

$$W \subset V$$

Linear Mapping

$$f: V \to W$$

(1) $f(\vec{x}+\vec{y}) = f(\vec{x}) + f(\vec{y})$

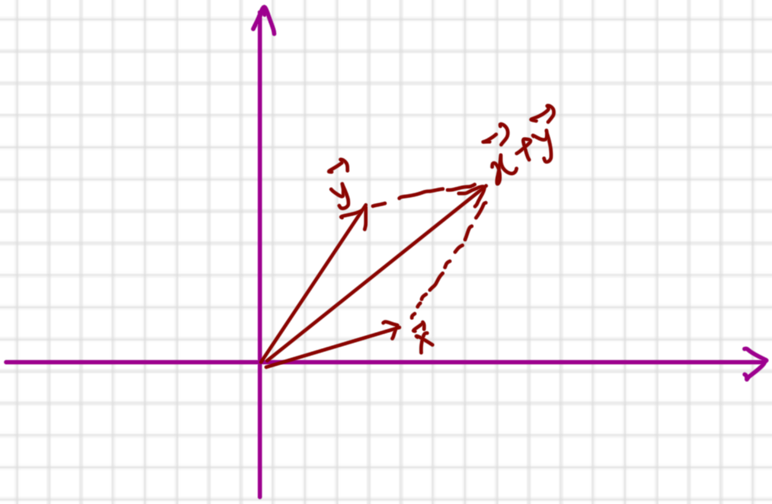
(2) $f(a\vec{x}) = af(\vec{x})$

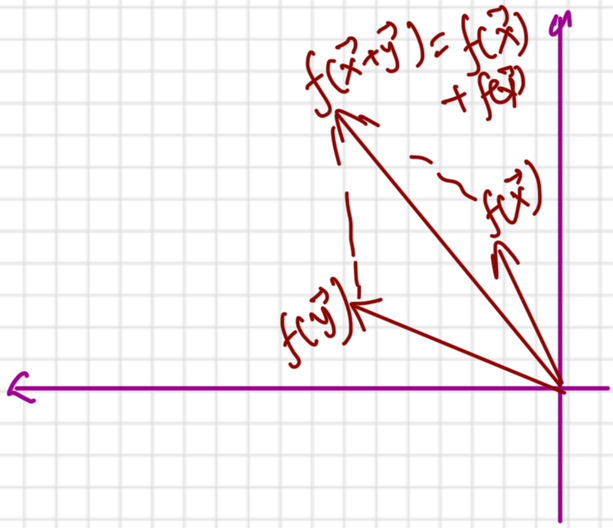
(3) $f(\vec{0}_V) = f(\vec{0}_W)$

$\vec{0}_V$ in V, $\vec{0}_W$ in W

(4) $f(-\vec{x}) = -f(\vec{x})$

(5) $f(a\vec{x}+b\vec{y}) = af(\vec{x}) + bf(\vec{y})$





Subspace Generation and Linear Dependence

$x_1\vec{a}_1 + x_2\vec{a}_2 + \cdots + x_n\vec{a}_n$ is a linear combination

$x_1, x_2 \ldots \in \mathbb{K}$

$\vec{a}_1, \vec{a}_2 \ldots \vec{a}_n \in V$

Let W be the set of all linear combinations of $\vec{a}_1, a_2 \cdots, \vec{a}_n$ then W be the smallest subspace of V containing

$$A = \{\vec{a}_1, \vec{a}_2 \ldots \vec{a}_n\}$$

We call this W the subspace generated or spanned by A which we can write as

$$\langle A \rangle = \langle \vec{a}_1, \vec{a}_2 \ldots \vec{a}_n \rangle$$

Finite-dimensional Vector Space

If V is generated by a finite number of vectors, then it is a finite-dimensional vector space.

It is important to find a set $\{\vec{a}, \vec{a}_2 \ldots \vec{a}_n\}$ that generates a finite-dimensional vector in the given vector space.

Even more, we are interested in finding smallest set i.e find smallest n.

Another point to note that all calculations related to linear algebra on V can be translated into the world of n-dimensional coordinate space $\mathbb{K}^n$.

Linear Dependence and Linear Independence

Let $A = \{\vec{a}_1, \vec{a}_2 \dots \vec{a}_n\} \subseteq V$, then we always have

$$0\vec{a}_1 + 0\vec{a}_2 + \dots + 0\vec{a}_n = \vec{0}$$

If
$$x_1\vec{a}_1 + x_2\vec{a}_2 + \dots + x_n\vec{a}_n = \vec{0} \quad \text{—} (1)$$
holds for some $x_1, x_2 \dots x_n \in \mathbb{K}$, then if, at least one of them is not 0, then we say that A is linearly dependent.

Example. $\{2\vec{a}, 3\vec{a}\}$ for any vector $\vec{a}$ is linearly dependent because, we can show that

$$3 \cdot 2\vec{a} + (-2) \cdot 3\vec{a} = \vec{0}$$

we chose this to show that a linear combination of them give zero vector.

On the other hand, we say that A is linearly independent if it is not linearly dependent, that is, eqn (1) holds only when $x_1 = x_2 = x_3 = \dots x_n = 0$

Example:

$A = \{(1,2), (2,3)\} \subseteq \mathbb{R}^2$ is linearly independent?

$$x\begin{bmatrix}1\\2\end{bmatrix} + y\begin{bmatrix}2\\3\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\begin{aligned} x + 2y &= 0 \\ 2x + 3y &= 0 \end{aligned} \quad \rightsquigarrow \quad \begin{aligned} \cancel{2x} + 4y &= 0 \\ \cancel{2x} + 3y &= 0 \\ \hline y &= 0 \\ x &= 0 \end{aligned}$$

Proves that A is linearly independent

In Python

```
>> from sympy import solve
>> from sympy.abc import x,y
>> ans = solve([x+2*y, 2*x +3*y], [x,y])
>> print(ans)
```

Example:

$A = \{(1,2), (2,4)\} \subseteq \mathbb{R}^2$

$$\begin{aligned} x + 2y &= 0 \\ 2x + 4y &= 0 \end{aligned}$$

$(x,y) = (2,-1)$ –

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

linearly independent

Summary:

Linear dependence of A is equivalent to "there exists a vector in A that is a linear Combination of other vectors of A".

velocity
acceleration

2 velocity + 3 acceleration $\}\rightarrow$ Redundant

BASIS and REPRESENTATION

Let V be a vector space (linear space) over a scalar field and

$$X = \{\vec{a}_1, \vec{a}_2 \, -- \, \vec{a}_n\} \subseteq V$$

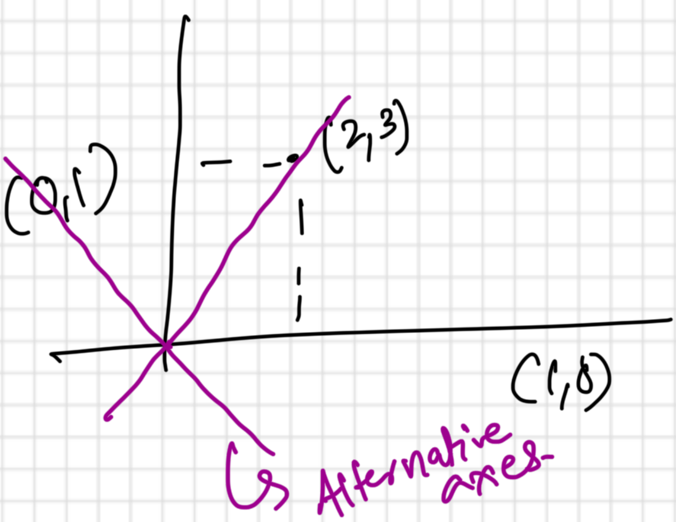
X is called basis of V, if it is linearly independent and generates V.

(0,1)
(1,0) $\}$ ← Basis in 2D.

(5,4)

$5(1,0) + 4(0,1)$

$= (5,4)$



Some Points to note:

① A set obtained by adding a new vector to X or removing any vector of X is no longer a basis of V.

$$\left((1,0,0), (0,1,0), (0,0,1) \right)$$

② A set obtained by replacing any vector of X with its nonzero scalar multiple remains a basis.

$$\left((1,0), (0,1) \right) \rightarrow \left((2,0), (0,1) \right)$$

③ A set obtained by replacing any vector of X with a sum of it and a scalar multiple of another vector of X remains a basis.

$\{\vec{e}_1, \vec{e}_2 \cdots \vec{e}_n\}$ is a standard basis where

$$\vec{e}_1 = (1, 0, \cdots 0)$$
$$\vec{e}_2 = (0, 1, \cdots 0)$$
$$\vdots$$
$$\vec{e}_n \; (0, 0 \cdots 1)$$

Let X be the basis of V.

Assume that we serialize the vectors in X as $\vec{a}_1, \vec{a}_2 \cdots \vec{a}_n$ and fix this order.

For any vector $\vec{x} \in V$

$$\vec{x} = x_1\vec{a}_1 + x_2\vec{a}_2 \cdots + x_n\vec{a}_n$$

$\vec{x} = x_1\vec{e}_1 + x_2\vec{e}_2 + \cdots + x_n\vec{e}_n$ in Standard basis.

The vector $\vec{x} = (x_1, x_2, \ldots x_n) \in \mathbb{K}^n$ made of the expansion coefficient $x_1, x_2, \ldots, x_n$ is called the representation of $\vec{x}$ on the basis X.

$$\vec{x} = x_1(1,0,0 \cdots 0) + x_2(0,1,0 \cdots 0) + \cdots$$
$$\cdots + x_n(0,0, \ldots 1)$$

Dimension of V

length of basis is the dimension

$$m = \dim_{\mathbb{K}} V \quad \text{or} \quad m = \dim V$$

$\mathbb{R}^n$ over $\mathbb{R}$

$\downarrow$

V.S.

$\mathbb{C}^n \quad \mathbb{R}^{2n}$

$\mathbb{C}^n$ over $\mathbb{R}$

$\downarrow$

Complex vector space

Rank

The dimension of the subspace generated by

$A = \{a_1, a_2 \cdots a_n\}$

is called the rank of A.

1. $\text{rank } A \leq n$
2. If A is linearly independent, then $A = n$
3. If A is linearly dependent, then $A < n$.

Direct Sum

$W_1, W_2 \cdots W_k$ subspaces

$$W = \{ \vec{x}_1 + \vec{x}_2 \cdots + \vec{x}_k \mid \vec{x}_1 \in W_1, \vec{x}_2 \in W_2, \ldots, \vec{x}_k \in W_k \}$$

then W is the direct sum

$$W = W_1 \oplus W_2 \oplus W_3 \oplus \cdots \oplus W_k$$

In Python, direct sum

```
>> [1,2] + [3,4,5]
[1,2,3,4,5]
```

Using numpy.

```
>> from numpy import array, concatenate
>> concatenate([array([1,2]),
                array([3,4,5])])
```

Using sympy

```
>> from sympy import Matrix
>> Matrix([1,2]).col_join(Matrix([3,4,5]))
   Output: Matrix([[1]
                   [2]
                   [3],
                   [4],
                   [5]])
```

We denote n-dimensional arrangement by arrays in Numpy.

① 1D arrangement is a sequence of elements arranged in a row.

② 2D arrangement is a matrix in which elements are arranged vertically and horizontally in 2D plane.

③ 3D arrangement is a layout of elements arranged vertically, horizontally and depthwise

```
>> from numpy import array
>> A = array( [1,2,3])        # 1D
>> B = array([ [1,2,3], [4,5,6]] )    # 2D
>> C = array( [[[1,2], [3,4]], [[5,6], [7,8]]] )
```

Vecter BroadCasting in Python

Vector broadcasting is purely computer op.

```
>> v = np.array( [[4,5,6]] ) # row vector
>> w = np.array( [[10,20,30]]).T  # Column vector.
```

$$\begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix} + \begin{bmatrix} 4, 5, 6 \end{bmatrix}$$

```
>> v+w
   array( [[14,15,16
            24,25,26
            34,35,36]])
```

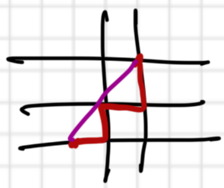
We are adding two vectors of dimension 1x3 and 3x1. Broadcasting is equivalent of repeating an operation multiple times between one vector and each element of another vector.

```
>> v= np.array([[1,2,3]]).T  # 3x1
>> w= np.array([[10,20]])  # 1x2
```

[1,1] + [10,20]
[2,2] + [10,20]
[3,3] + [10,20]

```
>> v+w
   array([[11, 21],
          [12, 22]
          [13 23]])
```

Broadcasting allows for compact and efficient calculations in numerical coding.



Vector Magnitude or Norm

$\vec{v} = [v_1, v_2, v_3]$

L_1 Norm: $||\vec{v}||_1 = |v_1| + |v_2| + |v_3|$ → Manhattan Distance

L_2 Norm: $||\vec{v}||_2 = \sqrt{v_1^2 + v_2^2 + v_3^2}$

= Euclidean distance

Max Norm: $||\vec{v}||_\infty = \max(|v_1|, |v_2|, |v_3|)$

L_p Norm: $||\vec{v}_p|| = (v_1^p + v_2^p + v_3^p)^{1/p}$

Unit Vector

$$\vec{v} = \frac{1}{\|\vec{v}\|_2}\vec{v}$$

Dot Product

$\vec{v} = [v_1, v_2, v_3]$

$\vec{w} = [w_1, w_2, w_3]$

$\vec{v} \cdot \vec{w}$ is the dot product defined as

$\vec{v} \cdot \vec{w} = v_1 w_1 + v_2 w_2 + v_3 v_3$ which is a scalar.

$$\vec{v} \cdot \vec{w} \in \mathbb{K}$$

Hadamard Product (Element wise Multiplication)

$$\begin{bmatrix} 5 \\ 4 \\ 8 \\ 2 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 0 \\ 0.5 \\ -1 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \\ 4 \\ -2 \end{bmatrix}$$

```
>> a = np.array([5, 4, 8, 2])
>> b = np.array([1, 0, 0.5, -0.1])
```

>> a*b # Hadamard product

Outer Product

The outer product is a way to create a matrix from a column vector and a row vector.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} d & e \end{bmatrix} = \begin{bmatrix} ad & ae \\ bd & be \\ cd & ce \end{bmatrix}$$

$\vec{w} = \begin{bmatrix} d \\ e \end{bmatrix}$

Outer product will be indicated as $\boxed{\vec{v} \cdot \vec{w}^T}$

$\vec{w}^T$ is the transpose of $\vec{w}$.

This notation makes an assumption that vectors are in column-wise orientation $\vec{v}$ means a column.

Note : $\vec{v}^T \vec{w}$ is the dot product.

Matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}_{m \times n}$$

n columns

m rows

a_{ij} = element at ith-row and jth-column

a_{ii} = Diagonal elements

$m = n \rightarrow$ Square matrix

Diagonal Matrix : A square matrix is a diagonal matrix if all its elements other than diagonal elements are zero.

In python:

```
>> from numpy import array
>> A = array([[1,2,3],[4,5,6]])
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

↳ A matrix of dimension 2×3

what data type: Tuple

```
>> A.reshape((3,2))
```

Matrix and Linear Mappings

we consider a general relation: $\vec{y} = \vec{f}(\vec{x})$

Let V and W be vector spaces over $\mathbb{K}$ of dimensions n and m.

$f: V \rightarrow W$ is the linear mappings.

$$V \rightarrow \boxed{\vec{f}} \rightarrow W$$

Let $X = \{\vec{v}_1, \vec{v}_2, \ldots \vec{v}_n\}$

$Y = \{\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_m\}$

So $\vec{f}(\vec{v}_1), \vec{f}(\vec{v}_2) \ldots \vec{f}(\vec{v}_n)$ are vectors in W

and Y is a basis of W.

So,

$$\vec{y}_1 = \vec{f}(\vec{v}_1) = a_{11}\vec{w}_1 + a_{21}\vec{w}_2 + \cdots + a_{m1}\vec{w}_m$$

$$\vec{y}_2 = \vec{f}(\vec{v}_2) = a_{12}\vec{w}_1 + a_{22}\vec{w}_2 + \cdots + a_{m2}\vec{w}_m$$

$$\vdots$$

$$\vec{y}_n = f(\vec{v}_n) = a_{1n}\vec{w}_1 + a_{2n}\vec{w}_2 + \cdots + a_{mn}\vec{w}_m$$

$a_{ij} \in \mathbb{K}$

In vector notation

$$\vec{a}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} \qquad \vec{a}_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix}$$

$$\vec{a}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

which are representation of $\vec{f}(\vec{v}_1), \vec{f}(\vec{v}_2)$
$\ldots f(\vec{v}_n)$

on basis Y.

For a vector $\vec{x} \in V$, let $\vec{y} = f(\vec{x})$

and $\vec{x}$ is n-dimensional such that $\vec{x} = (x_1, x_2 \ldots x_n)$

and $\vec{y}$ is m-dimensional such that

$$\vec{y} = (y_1, y_2 \dots y_m)$$

So in terms of basis:

$$\vec{x} = x_1\vec{v_1} + x_2\vec{v_2} + \dots + x_n\vec{v_n}$$
$$\vec{y} = y_1\vec{w_1} + y_2\vec{w_2} + \dots + y_m\vec{w_m}$$

Then we can write.

$$y_1\vec{w_1} + y_2\vec{w_2} + \dots y_m\vec{w_m} = \vec{f}(x_1\vec{v_1} + x_2\vec{v_2} \dots + x_n\vec{v_n})$$
$$= x_1\vec{f}(\vec{v_1}) + x_2 f(\vec{v_2}) + \dots + x_n f(\vec{v_n})$$
$$= (a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n)\vec{w_1}$$
$$+ (a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n)\vec{w_2}$$
$$+$$
$$\vdots$$
$$+ (a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn})\vec{w_m}$$

So by linear independence of γ, we can write

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \\ a_{21}x_2 + a_{22}x_2 + \dots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \end{bmatrix}$$

$$A = [\vec{a_1}, \vec{a_2} \dots \vec{a_n}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

then $A\vec{x}$ is defined as

$$A\vec{x} = x_1\vec{a}_1 + x_2\vec{a}_2 + \cdots + x_n\vec{a}_m$$

$$A\vec{x} = \underset{m \times n}{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & | & & | \\ & | & & | \\ a_{m1} & & & a_{mn} \end{bmatrix}} \underset{n \times 1}{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}$$

$$= \underset{m \times 1}{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}$$

which leads to $\vec{y} = A\vec{x}$

which is a matrix representation of $\vec{y} = \vec{f}(\vec{x})$

In python, a product of $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and a vector $\begin{bmatrix} 5 \\ 6 \end{bmatrix}$

```
>> A = array([[1,2],[3,4]])
>> A.dot([5,6])
```

$$\underset{2 \times 2}{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}} \times \underset{2 \times 1}{\begin{bmatrix} 5 \\ 6 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 \times 5 & 2 \times 6 \\ 3 \times 5 & 4 \times 6 \end{bmatrix} = \begin{bmatrix} 17 \\ 39 \end{bmatrix}$$

Example

$$A = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

→ A special matrix that rotates a vector on 2D plane by θ CCW.

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\vec{y} = A\vec{x} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\Rightarrow y_1 = x_1\cos\theta - x_2\sin\theta$$
$$y_2 = x_1\sin\theta + x_2\cos\theta$$

Matrix Product (Matrix-Matrix multiplication)

$$A_{m\times n} \times B_{n\times k} = C_{m\times k}.$$

$$\text{where } c_{ij} = \sum_{r=1}^{n} a_{ir} b_{rj}$$

a_{ir} = the element at ith-row and r-th column of A.

b_{rj} = the element at rth-row and jth column of B

Triangular Matrix

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

→ upper triangular matrix

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \\ 7 & 8 & 9 \end{bmatrix}$$

→ Lower triangular matrix

Identity Matrix

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Inverse of a Matrix

A^{-1} is the inverse of a matrix.
if $AA^{-1} = I = A^{-1}A$

① $(A^{-1})^{-1} = A$
② $(AB)^{-1} = B^{-1}A^{-1}$

A = Square matrix
B = Square matrix

But $AB = I$ are possible when A and B are not square.

Adjoint and Transpose of matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & \cdots & \cdots & a_{mn} \end{bmatrix}_{m \times n}$$

$$A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ \vdots & & & \vdots \\ a_{1n} & & \cdots & a_{mn} \end{bmatrix} \text{ : Transpose}$$

$$A^* = \begin{bmatrix} \overline{a_{11}} & \overline{a_{21}} & \cdots & \overline{a_{m1}} \\ \vdots & & & \vdots \\ \overline{a_{1n}} & \overline{a_{2n}} & \cdots & \overline{a_{mn}} \end{bmatrix} \text{ : adjoint matrix}$$

Here a_{ij} are complex numbers. $\overline{a_{ij}}$ are complex conjugate.

If a_{ij} are real, then

$$A^T = A^*$$

$$(A^{-1})^T = (A^T)^{-1}$$

Determinant

$|A|$: Determinant

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$|A| = ad - bc$$

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

$$|A| = a(ei - hf) - b(di - gf) + c(dh - eg)$$

For a large square matrix (4x4 or above) the determinant is computed using methods such as Laplace mechanism or LU decomposition.

```
>> A = np.array([[1,2], [3,4]])
>> det_A = np.linalg.det(A)
```

Calculating an inverse of a matrix

For a matrix A, its inverse is

$$A^{-1} = \frac{adj(A)}{det(A)} \text{ or } \frac{adj(A)}{|A|}$$

where adj(A) is the adjoint of a matrix A.
(not the same as adjoint matrix)

adj(A) is also called adjugate matrix.

Example:

$$A = \begin{bmatrix} -2 & 5 & 1 \\ 4 & 1 & 0 \\ -3 & 5 & 5 \end{bmatrix}$$

$$\text{adj}\, A = \begin{bmatrix} C_{11} & C_{21} & C_{31} \\ C_{12} & C_{22} & C_{23} \\ C_{13} & C_{23} & C_{33} \end{bmatrix}$$

C_{ij} are called cofactors.

$$C_{11} = + \left| \begin{bmatrix} 1 & 0 \\ 5 & 5 \end{bmatrix} \right| = 5 - 0 = 5$$

$$C_{12} = - \left| \begin{bmatrix} 4 & 0 \\ -3 & 5 \end{bmatrix} \right| = -(20-0) = -20$$

$$C_{13} = + \left| \begin{bmatrix} 4 & 1 \\ -3 & 5 \end{bmatrix} \right| = 20+3 = 23$$

Similarly for other C_{ij}

$$\text{adj}\, A = \begin{bmatrix} 5 & -20 & -1 \\ -20 & -7 & 4 \\ 23 & -5 & -22 \end{bmatrix}$$

General formula

(I) first, Calculate the cofactors

$C_{ij} = (-1)^{i+j} M_{ij}$ where M_{ij} is the determinant of $(n-1) \times (n-1)$ matrix resulting from deleting row i and column j of A.

② Divided Co factor matrix by det (A).

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

$$A^{-1}\vec{y} = A^{-1}A\vec{x}$$

$$A^{-1}\vec{y} = I\vec{x}$$

Rank of a Matrix

$$A = [\vec{a}_1, \vec{a}_2 - - \cdot \ \vec{a}_n]_{m \times n}$$

The rank of the matrix is the rank of set $\{\vec{a}_1, \vec{a}_2 \cdots \vec{a}_n\}$ or in other words, the max. no. of linearly independent row or column vectors in the matrix.