CPE 490 590 Spring 2025

Instructor: Rahul Bhadani
The University of Alabama in Huntsville

Linear Algebra Review

Linear Combination: $\sum_{i}^{n} w_i x_i$

x_i = feature
w_i = weight of each feature

A system of linear equations:

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

Vector:

Coloumn vector

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \text{ or } \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

How will you write coloumn vector in python?

How many rows and columns it has?

Bold face

Row vector $\vec{v} = [v_1 \quad v_2 \quad v_3]$ or $\mathbf{v} = [v_1 \quad v_2 \quad v_3]$

How will you write row vector in python?

How many rows and coloumns it has?

Scalar field A function that assigns a single number to each point in a region of space or space time.

E.g. Temperature distribution is space.

Space or Space time can be 2D, 3D or even higher dimension.

Another example : $x^2+y^2+z^2$ gives one value

So $f(x,y,z) = x^2+y^2+z^2$ is a scalar field.

We can denote a scalar field by $\mathbb{K}$

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

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

   Closed under summation

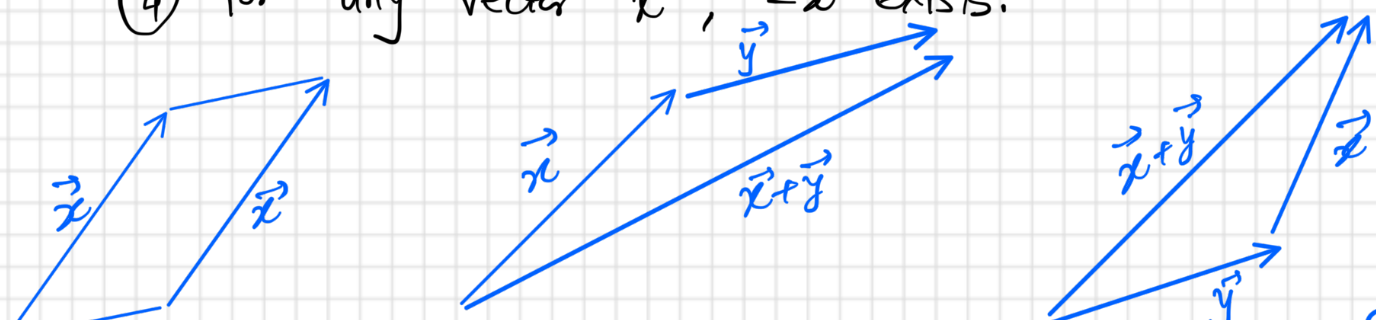
2. For any vector $\vec{x}$ and a scalar a, $a\vec{x} \in V$

   What happens to the vector $\vec{x}$ when a is negative?

   Closed under scalar multiplication

3. There exists a zero vector $\vec{0}$.

4. For any vector $\vec{x}$, $-\vec{x}$ exists.



Vector Sum

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ \vdots \\ x_n+y_n \end{bmatrix} = \begin{bmatrix} y_1+x_1 \\ y_2+x_1 \\ \vdots \\ y_n+x_n \end{bmatrix}$$

Subspaces

Let V be a vector space over $\mathbb{K}$

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

① $\vec{x}+\vec{y} \in W$ for any $\vec{x}, \vec{y} \in W$ (closed under vector sum)

② $a\vec{x} \in W$ $\quad a \in \mathbb{K}$ and $\vec{x} \in W$

(closed under scalar multiplication)

③ $\vec{0} \in W$

④ $-\vec{x} \in W$ for any $\vec{x} \in W$

A subspace is entirely contained within another vector space.

So $\quad W \subset V \quad$ (W is a subset of V)

Linear Mappings

A mapping $f: V \to W$ is called linear mapping if it reflects upon W the structure of V consisting of the vector sum, scalar multiple, zero vector and inverse vector, that is

① $f(\vec{x}+\vec{y}) = f(\vec{x}) + f(\vec{y})$ for and $\vec{x}, \vec{y} \in V$

② $f(a\vec{x}) = a f(\vec{x})$

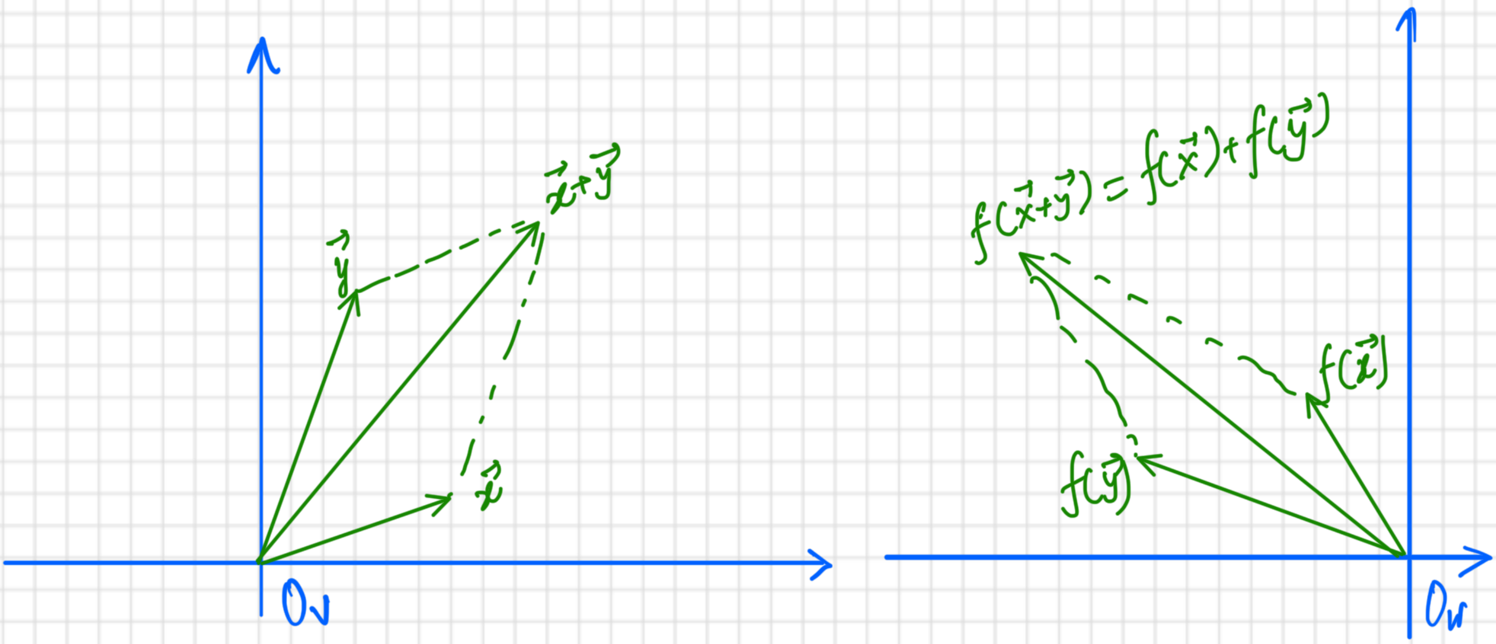
③ $f(\vec{O}_V) = f(\vec{O}_W)$

$\vec{O}_V$ means zero vector in V

$\vec{O}_W$ means zero vector in W.

④ $f(-\vec{x}) = f(\vec{x})$

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



SUBSPACE GENERATION AND LINEAR INDEPENDENCE

Earlier we saw linear combination that we can rewrite as

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

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

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

Let W be the set of all linear combinations of $\vec{a}_1, \vec{a}_2 \cdots \vec{a}_n$

then W becomes the smallest subspace of V containing

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

We call this W the subspace generated by (or spanned by) A can we can alternatively write

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

A trivial subspace $\{\vec{0}\}$ is the subspace of V generated by empty set $\{\}$

Finite-dimensional Vector Space:

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

It is important to find a set $\{\vec{a}_1, \vec{a}_2 \cdots \vec{a}_n\}$ that generates a finite-dimensional vector.

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 the 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

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

If $x_1\vec{a}_1 + x_2\vec{a}_2 + \cdots + x_n\vec{a}_n = \vec{0}$ —— (1)

holds for some $x_1, x_2, \dots x_n \in \mathbb{K}$ at least one of them is not 0, we say that A is linearly dependent.

Example:

$\{2\vec{a}, 3\vec{a}\}$ for any vector $\vec{a}$ is linearly dependent because

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

We choose this to show that a linear combination of them may 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 $x_1 = x_2 = \cdots x_n = 0$

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

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

holds for $x, y \in \mathbb{K}$, that is

$$x + 2y = 0$$
$$2x + 3y = 0$$

The system has a unique solution $x = y = 0$.
Hence we see 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:
$$A = \{(1,2), (2,4)\} \subseteq \mathbb{R}^2$$

$$x + 2y = 0$$
$$2x + 4y = 0$$

which has the solution $(x,y) = (2,-1)$
other than $(x,y) = (0,0)$

Hence A is not linearly dependent.

Homework Question Start with

$A = \{\overset{x}{(1,2,3)}, \overset{y}{(2,3,4)}, \overset{z}{(3,4,5)}\} \longrightarrow x + 2y + 3z = 0 \ldots$

$B = \{\underset{x}{(1,2,3)}, \underset{y}{(2,3,1)}, \underset{z}{(3,1,2)}\} \longrightarrow x + 2y + 3z = 0 \ldots$

Write the system of equations to test the linear independence of A and B and solve them using sympy package from Python.

To summarize, linear dependence of A is equivalent to "there exists a vector in A that is a linear combination of the other vectors of A". The linear independence of A is equivalent to "any vecter in A never belongs to the subspace generated by the other vecter but it".

BASIS and REPRESENTATION

Basis Let V be a linear space over a scalar field and $X = \{\vec{a}_1, \vec{a}_2, \dots \vec{a}_n\} \subseteq V$

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

Some points to note:

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

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

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

Standard basis

The set $\{\vec{e}_1, \vec{e}_2 \dots, \vec{e}_n\}$ is given by

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

is a bases of $\mathbb{K}^n$. It is called as the standard basis.

Let X be the basis of V.

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

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

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

We call this the expansion of $\vec{x}$ on the basis of X.

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

With the standard basis, we could write

$$\vec{x} = x_1\vec{e}_1 + x_2\vec{e}_2 + \dots + x_n\vec{e}_n$$

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

(Note that at this point we are currently not caring about whether it is a column vector or row vector)

$s\vec{x} + t\vec{y}$ implies $f_X(s\vec{x} + t\vec{y})$ where f_X is bijective mapping from V to $\mathbb{K}^n$

with

$$f_X(s\vec{x} + t\vec{y}) = s f_X(\vec{x}) + t f_X(\vec{y})$$

Bijective mapping refreshes:

- Every element in $\mathbb{K}^n$ is mapped to by exactly one element in V.
- Each element in V maps to a unique element in $\mathbb{K}^n$.

Dimension of V.

If V has a basis with m vectors, any basis of V consists of m vectors. We call this m the dimension of V over $\mathbb{K}$ and $m = \dim_{\mathbb{K}} V$

Q: What is the dimension of a vector space $\mathbb{R}^n$ over $\mathbb{R}$?

What is the dimension of a vector space $\mathbb{C}^n$ over $\mathbb{R}$?

Rank

The dimension of the subspace generated by $A = \{a_1, a_2 \dots, a_n\}$ is called the rank of A. We denote it by $\text{rank}_{\mathbb{K}} A$ or $\text{rank}\, A$.

Some Points:

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

If A is a subset of an m-dimensional vector space V, then

1. $\text{rank}\, A \leq m$
2. If A generates V, then $\text{rank}\, A = m$.
3. If A doesn't generate V, then $\text{rank}\, A < m$.

Direct Sum

Let $W_1, W_2 \dots W_k$ be subspaces of vector space V.

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

then W is a subspace of V.

W is called as sum of subspaces.

In this case, when every element $\vec{x}$ of W is uniquely expressed $\vec{x} = \vec{x}_1 + \vec{x}_2 + \dots + \vec{x}_k$ with $\vec{x}_1 \in W_1$, $x_2 \in W_2 \dots$ then we call W the direct sum of $W_1, W_2 \dots W_k$

We can also write $W = \{W_1, W_2 \dots W_k\}$ and denote the direct sum as $W_1 \oplus W_2 \oplus \dots \oplus W_k$

In python: direct sum can be written as

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

output: `[1,2,3,4,5]`

or using numpy

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

In sympy:

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

output:

```
Matrix([ [1],
         [2],
         [3],
         [4],
         [5]])
```

```
Matrix([[1,2]])
```

We denote n-dimensional arrangement by arrays in Numpy Python.

1. 1-D 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 a 2D plane
3. 3D arrangement is a layout of elements arranged vertically, horizontally, and depth-wise in 3D space.

```
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]]])
```

```
A= [1,2,3]
B=  [[1,2,3]
     [4,5,6]]        — 2×3 matrix

C=  [[ [1,2]
       [3 4]]        — 2 2D arrays
     [ [5,6]
       [7 8]]]
```

Vector Broadcasting in Python

Vector broadcasting is purely a computer operation.

Consider Python code:

```
>> v= np.array( [[4,5,6]])  # row vector (3 columns)
>> w= np.array( [[10,20,30]]).T   # column vector. (3 rows)
>> v+w
   array( [[14, 15, 16]
           [24, 25, 26]
           [34, 35, 36]])
```

What is going on?

We are adding two vectors of dimension 1x3 and 3x1. Clearly, there is a dimension mismatch but there doesn't seem to be an error.

Here, broadcasting operation is taking place even though there is a dimension mismatch.

Broadcasting essentially means to repeat an operation multiple times between one vector and each element of another vector.

Consider:

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

Then it does the following operation on v+w

[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

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

L1 Norm: $\|\vec{v}\|_1 = |v_1| + |v_2| + |v_3|$

L2 Norm: $\|\vec{v}\|_2 = \sqrt{v_1^2 + v_2^2 + v_3^2}$ → Magnitude

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

Creating a unit vector

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

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

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

$\vec{v} \cdot \vec{w} = v_1 w_1 + v_2 w_2 + v_3 w_3$ which is a scalar.

Hadamard Product:

This is just fancy way to call element-wise multiplication:

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

```
>> a= np.array([5,4,8,2])
>> b= np.array([1, 0, 0.5, -1.0])
>> a*b      # Hadamard product
```

what would happen if b= np.array([1,0]) ?

Outer Product

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

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

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

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

In this case $\vec{v}^T\vec{w}$ would be the dot product.

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

MATRIX

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}_{m \times n} \rightarrow \begin{array}{l} m \text{ rows} \\ n \text{ columns.} \end{array}$$

a_{ij} = element at i-th row and j-th column.

a_{ii} = Diagonal elements

$m = n \rightarrow$ Square Matrix

Diagnal 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]])
```

→ what is its dimension?
2×3.

Reshaping to change the dimension:

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

Output:

```
array([[1,2],
       [3,4],
       [5,6]])
```

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. , and $f: V \to W$ is the linear mapping.

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

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

$\vec{f}(\vec{v}_1), \vec{f}(\vec{v}_2) \dots \vec{f}(\vec{v}_n)$ are vectors in W, and Y is a basis of W.

So

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

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

$$\vdots$$

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

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

In vector notation, we can further write:

$$\vec{a}_1 = \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{bmatrix} \qquad \vec{a}_2 = \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{m2} \end{bmatrix} \quad \dots \quad \vec{a}_n = \begin{bmatrix} a_{1n} \\ a_{2n} \\ \vdots \\ a_{mn} \end{bmatrix}$$

which are representation of $\vec{f}(\vec{v}_1), \vec{f}(\vec{v}_2) \dots, \vec{f}(\vec{v}_n)$ on basis Y.

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

and $\vec{x}$ is n-dimensional such that

$$\vec{x} = (x_1, x_2, \dots x_n)$$

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 could write

$$y_1 \vec{w}_1 + y_2 \vec{w}_2 + \dots + y_m \vec{w}_m = \vec{f}(x_1 \vec{v}_1 + x_2 \vec{v}_2 + \dots + x_n \vec{v}_n)$$

$$= x_1 \vec{f}(\vec{v}_1) + x_2 \vec{f}(\vec{v}_2) + \dots + x_n \vec{f}(\vec{v}_n)$$

$$= (a_{11} x_1 + a_{12} x_n + \dots + a_{1n} x_n) \vec{w}_1$$

$$+ (a_{21} x_1 + a_{22} x_2 + \dots + a_{2n} x_n) \vec{w}_2$$

$$+ \\ \vdots$$

$$+ (a_{m1} x_1 + a_{m2} x_2 + \dots + a_{mn} x_n) \vec{w}_m$$

By linear independence of γ, we can write

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

So if we define a matrix A such that

$$A = [\vec{a}_1, \vec{a}_2 \dots \vec{a}_n]$$

then

$A\vec{x}$ is defined as

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

$$A = [\vec{a}_1 \ \vec{a}_2 \cdots \vec{a}_n] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix}_{m\times n}$$

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}$$

$$A\vec{x} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & & \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix}_{m\times n}}_{A} \overbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}}^{\vec{x}}$$

$$= \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}_{m\times 1}}_{\vec{y}}$$

which leads

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

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

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

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

Answer: array ([17,39])

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

$$= \begin{bmatrix} 1\times 5 + 2\times 6 \\ 3\times 5 + 4\times 6 \end{bmatrix} = \begin{bmatrix} 5+12 \\ 15+24 \end{bmatrix} = \begin{bmatrix} 17 \\ 39 \end{bmatrix}$$

Example:

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

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

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

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

Question: what is special about A?

→ Rotates a vecter (x_1, x_2) on 2D plane by θ degrees CCW.

Matrix Product or Matrix-Matrix Multiplication

If a matrix A is $m \times n$
matrix B is $n \times k$
then their product is $m \times k$.

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

where elements of $C_{m\times k}$ is

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

a_{ir} = the element at i-th row and r-th column of A.
b_{rj} = the element at r-th row and j-th column of B.

```
>> A = np.array([[1,2], [3,4]])
>> B = np.array([[5,6], [7,8]])
>> C = A.dot(B)
```

C : [[19 22
43 50]]

Triangular Matrix

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix} \longrightarrow \text{upper triangular matrix}$$

$$U = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 5 & 0 \\ 7 & 8 & 9 \end{bmatrix} \longrightarrow \text{lower triangular matrix}$$

Identity Matrix

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

Diagonal Matrix

$$D = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 23 \end{bmatrix}$$

Inverse of a Matrix

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

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

However, $AB = I$ is possible for non-square matrices A and B as well.

Homework:

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

$B = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix}$

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

find $a, b, c, d, e,$ and f using sympy Python package

Adjoint and Transpose Matrix

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

then $A^T = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$

is the transpose of A.

Consider that a_{ij} are complex number in general.

the $A^* = \begin{bmatrix} \overline{a_{11}} & \overline{a_{21}} & \cdots & \overline{a_{m1}} \\ \overline{a_{12}} & \overline{a_{22}} & \cdots & \overline{a_{m2}} \\ \vdots & & \ddots & \vdots \\ \overline{a_{1n}} & \overline{a_{2n}} & \cdots & \overline{a_{mn}} \end{bmatrix}$

if a_{ij} are real numbers

$$A^T = A^*$$

$\overline{a_{11}}$ is the complex conjugate of a_{11}

A^* is called adjoint matrix of A, (also known as conjugate transpose, or Hermitian transpose)

For a square matrix A.

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

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

Rank of a matrix

$A = [\vec{a}_1 \ \vec{a}_2 \ \cdots \vec{a}_n]$ of shape (m, n)

where $\{\vec{a}_1, \vec{a}_2 \cdots \vec{a}_n\}$ are the set of vectors obtained by taking out all of the column vectors of A.

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 maximum number of linearly independent row or column vectors in the matrix.

→ It represents the dimension of the vector space spanned by the matrix's columns or rows.

or it is the dimension of the space spanned by the vectors a matrix contains (because a matrix can be thought to be a collection of vectors).

1. A rank of a matrix cannot exceed the number of rows or column of a matrix.
2. For a matrix of size (m, n) its rank is less than or equal to $\min(m, n)$.

Why we need to know the rank of a matrix

1. To determine if a system of linear equations has a unique solution or not.

② For dimensionality reduction or data compression etc.

Determinant

A determinant of a matrix gives a scalar value.

$|A|$ is the determinant of a matrix A.

consider a 2x2 matrix

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

$$|A| = ad - bc$$

For a 3x3 matrix $A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$

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

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

In python:

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

Trace of a Matrix :

Sum of all diagonal elements of A.

Denoted as $Tr(A)$.

Calculating an inverse of a matrix

For a matrix A, its inverse

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

where adj (A) is the adjoint of a matrix A. (this is different from Adjoint matrix A^* we saw earlier)

Adjoint of a matrix, also called Adjugate Matrix is calculated as follows:

Consider an example

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

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

where C_{ij} are called cofactors computed as

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

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

$$C_{13} = +\left|\begin{bmatrix}4 & 1\\ -3 & 5\end{bmatrix}\right| = 20+3 = 23 \qquad \begin{bmatrix}-2 & 5 & 1\\ 4 & 1 & 0\\ -3 & 5 & 5\end{bmatrix}$$

$$C_{21} = -\left|\begin{bmatrix}5 & 1\\ 5 & 5\end{bmatrix}\right| = -(25-5) = -20$$

$$C_{22} = +\left|\begin{bmatrix}-2 & 1\\ -3 & 5\end{bmatrix}\right| = -10(-(-3)) = -7$$

$$C_{23} = -\left|\begin{bmatrix}-2 & 5\\ -3 & 5\end{bmatrix}\right| = -(-10-(-15)) = -5$$

$$C_{31} = +\left|\begin{bmatrix}5 & 1\\ 1 & 0\end{bmatrix}\right| = +(0-1) = -1$$

$$C_{32} = +\left|\begin{bmatrix}-2 & 1\\ 4 & 0\end{bmatrix}\right| = -(0-4) = +4$$

$$C_{33} = +\left|\begin{bmatrix}-2 & 5\\ 4 & 1\end{bmatrix}\right| = -2-20 = -22$$

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

General formula:

① first Calculate the Co factors

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