## * What is eigen Value & eigen vector ?

⟹ Imagine you have big box called a Matrix.
when you put some (vectors) arrows into the box, they com out differently may be bigger Smaller or pointing in a different direction.

⟹ Now, There are some special arrows that even after going through box still point in $\boxed{\text{Same direction}}$. They might be longler or shorter, but they don't change which they are pointing. These special arrows called as $\boxed{\text{eigen vectors}}$.

⟹ An $\boxed{\text{eigenValue}}$ is like a magic Number attached to these special Arrows. It tells you how much these arrows stretch or shrink when they go through the box.

* eigen vector :- Special Arrows that don't change direction when you put them through a Matrix.

* eigen Values : How ~~th~~ much these special Arrows change in size.

⟹ EigenValues and eigenVectors help us understand how matrices transform things, which really useful in figuring out pattern in data.

## * . What is Significance of eigenValues and eigenVectors ?

⟹ In ML

### ① principal component Analysis (PCA)

⟹ EVal and EVec are fundamental to PCA, a popular dimensionality reduction technique.

⟹ Evec of Covariance Matrix represent the principle components which capture the direction of Maximum Variance in Data.

⟹ The corresponding eigenValue indicate the amount of variance explained by each principle components.

⟹ By selecting a subset of principle components with largest eigenvalues, one can reduce the dimensionality of data while preserving most of its variance.

### ② Linear Transformation

⟹ In ML data is transformed in to different spaces using linear transformation.

⟹ Eigenvectors provide basis for these transformation

⟹ They represent the direction in which the transformation stretches or compress the data. while the corresponding eigenvalue indicate scale of each vector.

### ③ Kernal Methods :

⟹ In kernal methods like SVM and kernal PCA, the data is simply mapped to highe dimensional space using a kernal function.

⟹ The eigenvector of the kernal Matrix provide the representation of data in higher dimension space.

→ **EVD and SVD both are matrix decomposition technique. but they serve different purposes and different Mathematical properties.**

| EVD | SVD |
|---|---|
| → Applicable to $\boxed{\text{Square Matrix}}$ | → Applicable to $\boxed{\text{Any matrix}}$ |
| → decomposes square matrix into a set of eigenvalues and eigenvectors. | → decomposes a matrix into 3 other matrices. $U, \Sigma, V^*$ Here $U, V^*$ = Unitary Matrix; $\Sigma$ = diagonal matrix containing singular value |
| → For square Matrix A $AV = \lambda V$ where $V$ - eigenvector, $\lambda$ - eigenValue | |
| → EVD is not defined for all matrices. its only applicable for square Matrix. | → SVD is exists for any matrix even it's not a square or invertible. |

# * Application of EVD or SVD

① PCA

→ Both SVD and EVD are used in PCA, a dimensionality reduction technique widely used in ML

② matrix factorization

→ SVD is commonly used in collaborative filtering Algorithm for RS.

→ It's used to factorize user-item interection matrix to lower-dimensional matrices representing users and items, capturing latent features

③ Image compression.

→ SVD is also used in image compression technique like the singular value decomposition based image compression.

# *· Eigen Values and Eigen Vectors

$$Ax = \lambda x \qquad \lambda \in \mathbb{R} \qquad x \neq 0$$

→ Any square matrix A and eigen vector is any vector X which is not equal to zero such that multiplying matrix A by vector X gives back some multiple of vector X that multiplier is lambda which can be any real number.

$$\boxed{\text{eigenvector} * \text{corresponding eigenvalue} = \text{original matrix} * \text{eigenvector}}$$

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

$$Ax = \lambda x = \lambda I x \qquad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$Ax - \lambda I x = 0$$

$$\underbrace{(A - \lambda I)}_{M} x = 0 \qquad \text{here } x \neq 0$$

here M is not invertible so it $\det(M) = 0$

$$\det(A - \lambda I) = 0$$

$$\det\left(\begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} - \lambda \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right)$$

$$\det\begin{pmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{pmatrix}$$

$$3\lambda + \lambda^2 + 2 = 0$$

$$\boxed{\lambda = -2, -1}$$

Now, Correspondin eigen Vector of $\lambda = -1$

$Ax = \lambda x$

$$\begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \overset{(-1)}{\begin{pmatrix} -x_1 \\ -x_2 \end{pmatrix}}$$

$\boxed{x_2 = -x_1}$ — (1)

$-2x_1 - 3x_2 = -x_2$ — (2)

$-2x_1 = 2x_2$

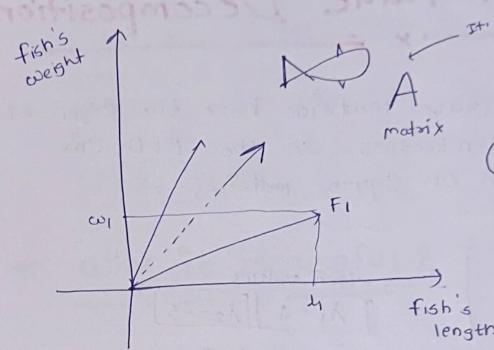
$\boxed{x_1 = -x_2}$ → So you can take any eigen Vector such that ($x_1 = -x_2$)

$$\begin{array}{cccc} A & x & \lambda & x \\ \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix} & \begin{pmatrix} 1 \\ -1 \end{pmatrix} = & -1 & \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{array}$$

$$\begin{pmatrix} -1 \\ -2+3 \end{pmatrix} = \begin{pmatrix} \cancel{-1} \\ +1 \end{pmatrix}$$

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ +1 \end{pmatrix}$$ hence prooved

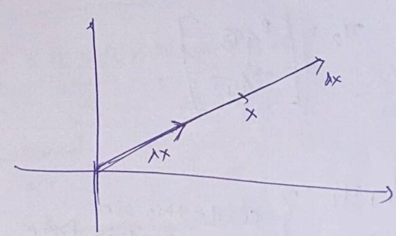
## * why it is used in Data science ?:




It is just linear transformation

$A \cdot F_1$ than we get new fish $f_2$

→ linear transformation represent given a fish a map ~~set~~ to the length and weight of the best friend of that fish.




⟹ If you know the eigenVector for a given matrix for given linear transformation.

⟹ You know that the linear transformation will map that eigenvector onto a different vector which maintains the same ratio.

# Eigen Value Decomposition

⟹ Only square matrice have the eigen-Values and eigen Vectors so the EVD can perform on square matrices.

$$A = \begin{bmatrix} 1 & 4 \\ 9 & 1 \end{bmatrix} \longrightarrow \text{eigen values } \boxed{\lambda_1 = 7} \boxed{\lambda_2 = -5}$$

→ corresponding to 7 $v_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

→ corresponding to -5 $v_2 = \begin{bmatrix} 2 \\ -3 \end{bmatrix}$

⟹ Now, we Normalized given vector.

$$u_1 = \begin{bmatrix} 2/\sqrt{13} \\ 3/\sqrt{13} \end{bmatrix} \qquad u_2 = \begin{bmatrix} 2/\sqrt{13} \\ -3/\sqrt{13} \end{bmatrix}$$

⟹ $\left.\begin{aligned} A u_1 &= \lambda_1 u_1 \\ A u_2 &= \lambda_2 u_2 \end{aligned}\right\}$ write this in matrix form

$$A \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

$$Au = u\Lambda$$

→ Take ~~it~~ $u^{-1}$ both side

$$\boxed{A = \underbrace{u}_{\text{contains eigen vector}} \underbrace{\Lambda}_{\text{contains eigen values}} u^{-1}} \rightarrow \text{eigen decomposition of } A$$

## why it is useful?

⟹ Consider $A^P$, (here Matrix is linear transformation of vector)

if $P = 8$

| without eigen decomp. | with eigendecomp |
|---|---|
| $\underbrace{\underbrace{\underbrace{A \times A}_{A^2} \times \underbrace{A \times A}_{A^2}}_{A^4} \times \underbrace{\underbrace{A \times A}_{A^2} \times \underbrace{A \times A}_{A^2}}_{A^4}}_{A^8}$ | $A^P = \underbrace{U \Lambda U^{-1}}_{A} \times \underbrace{U \Lambda U^{-1}}_{A} \ldots \underbrace{U\Lambda U^{-1}}_{A}$ ($U^{-1}U = I$) |
| Requires $\log_2(P) = \log_2 8 = 3$ Matrix ~~mult~~ moltiplication | $= u \Lambda^P u^{-1}$ |
| | $\Lambda^P$ = is very easy to compute because it is diagonal matrix |
| | $\Lambda^P = \begin{bmatrix} \lambda_1^P & \cdots & 0 \\ 0 & \lambda_2^P & 0 \\ \vdots & & \ddots \\ 0 & & \lambda_n^P \end{bmatrix}$ |
| | Requires 2 |

# * SVD

→ Any Matrix $A_{m \times n}$ divide into 2 unitary matrices which are orthogonal in nature and rectangle diagonale matrix ~~~~ of Singular value.

$$A_{m\times n} = U_{m\times m} \, \Sigma_{m\times n} \, V^T_{n\times n}$$

$U_{m\times m}$: orthogonal, left singular vector

$V^T_{n\times n}$: orthogona, Right singular Vector.

→ matrix A can be decompose in 2 ways

| | |
|---|---|
| → $U_{n\times n}$ | $U_{m\times d}$ |
| $\Sigma_{n\times d}$ | $\Sigma_{d\times d}$ |
| $V_{d\times d}$ | $V_{d\times d}$ |
| | preferable |
| $\boxed{\text{Full SVD}}$ | $\boxed{\text{Reduced SVD}}$ |

For this raw should be orthogonal

$\Sigma$ is diagonal Matraix

Here $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ (decreasing order)

where $\sigma_i$'s are Singular Values (Non negative + Real)

$$A_{m\times n} = U_{m\times m} \, \Sigma_{m\times n} \, V^T_{n\times n} = \Sigma_i \sigma_i u_i v_i^T$$

Scalar, Vector

# * Rank of Matrix

⟹ No of linearly independent columns are the $\boxed{\text{Rank of Matrices}}$

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} = 2$$

$$\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} = 1$$

→ because it is linear combinations of colum 1

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

$$\begin{array}{ccccc} & & ✓ & ✓ & ✓ \\ \times & \times & & & \end{array}$$

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

$$\begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \end{array}$$

$a_1 = \underline{1} a_1 + \underline{0} a_2$

$a_2 = \underline{\quad} a_1 + \underline{0} a_2$

$a_3 = \underline{1} a_1 + \underline{1} a_2$

$a_4 = \underline{1} a_1 + \underline{3} a_2$

$a_5 = \underline{2} a_1 + \underline{0} a_2$

linearly independent colum = 2

$\boxed{\text{Rank} = 2}$

$$\begin{bmatrix} | & | & | & | & | \\ a_1 & a_2 & a_3 & a_4 & a_5 \\ | & | & | & | & | \end{bmatrix} = \begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 1 & 2 \\ 0 & 1 & 1 & 3 & 0 \end{bmatrix}$$

Product $4\times5$ = $A''$ Ingrediats $4\times2$ · $R$ recipes $2\times5$

$$\underset{\substack{20 \\ \text{numbers} \\ \text{represent } A}}{A} = \underbrace{\underset{\substack{8 \\ 4\times2}}{A'} \times \underset{2\times5}{R}}_{\underbrace{8+10}_{18}}$$

$A = N \times P$

$A' = N \times K$ (K → Rank of matrix)

$R = K \times P$ (P → No. of original columns)

$K(N+P)$ +ion

* Compression Ratio :-

$$\frac{K(N+P)}{N\times P} = \boxed{K/P + K/N}$$

If $N = 1000$ → Data points, $P = 10$ → features, $K = 5$ → Rank of matrix

$$\text{Compression} = \frac{5}{10} + \frac{5}{1000} \quad \frac{505}{1000} \approx \boxed{50.5\%}$$

(100)

⇝ Only 5 of 10 original column needed ~~are~~

⇒ If we have large dataset and writing this like rank format than store in very small numbers.

**Rank:** $\mathbb{R}^{m\times n}$ (m ↑ Row, n ↑ col) $\boxed{\text{Rank } p \le n}$

So than we can express matrix small Rank matrices

$$\boxed{M = \sigma_1 \underbrace{u_1 v_1^T}_{m\times n} + \dots \sigma_p \underbrace{u_p v_p^T}}$$

Rank 1 matrices

$u_i = \begin{bmatrix} u_1 & u_2 & u_3 \\ | & | & | \end{bmatrix}_3$ $v_i \begin{bmatrix} v_1 & v_2 \end{bmatrix}_2$ $3\times2$

$$M = \underbrace{\begin{bmatrix} | & & | \\ u_1 & \cdots & u_p \\ | & & | \end{bmatrix}}_{\substack{m\times p \\ \text{Orthonormal}}} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_p \end{bmatrix}}_{\substack{p\times p \\ \text{diagonal}}} \underbrace{\begin{bmatrix} — v_1^T — \\ \vdots \\ — v_p^T — \end{bmatrix}}_{\substack{p\times n \\ \text{orthonormal}}}$$

$$\boxed{M = U \Sigma V^T} \quad (1)$$

⇒ Multiply eq (1) with $V$ (Right)

$\boxed{MV = U\Sigma}$ $(\because V^T V = I)$

$V_1 \dots V_p$ are Right singular vectors

⇒ Multiply eq (1) with $U$ (left

$\boxed{U^T M = \Sigma V^T}$

$U_1 \dots U_p$ are left singular vectors

$\sigma_1 \dots \sigma_p$ are singular values.

## * Why SVD is very useful ?

$\Longrightarrow$ Suppose $M \in \mathbb{R}^{\overset{m}{100} \times \overset{n}{10}}$ (100 ↓ rows, 10 features) $\Longrightarrow$ 1000 Numbers Needed

$\Longrightarrow$ Suppose $\sigma_1 = 3$, $\sigma_2 = 2$

$\boxed{\forall\, i > 2 \quad \sigma_i \simeq 0}$ other singular values are zero

So $M = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots 0$

here ($p=2$)

$$= \underbrace{\begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix}}_{\substack{100\times 2 \\ m\times p \\ 200}} \underbrace{\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_1 \end{bmatrix}}_{\substack{2\times 2 \\ p\times p \\ 4}} \underbrace{\begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \end{bmatrix}}_{\substack{2\times 10 \\ p\times n \\ 20}}$$

$\boxed{\text{224 Numbers Needed}}$

$\boxed{< 25\% \text{ reducted data Needed}}$

## * Analytically Method

$A = U \Sigma V^T$ where $U$ and $V$ are Orthnormal vector.

$UU^T = I$

$VV^T = I$

$\rightarrow$ For Finding $V$.

$A^T A = (V\Sigma^T U^T)(U \Sigma V^T)$

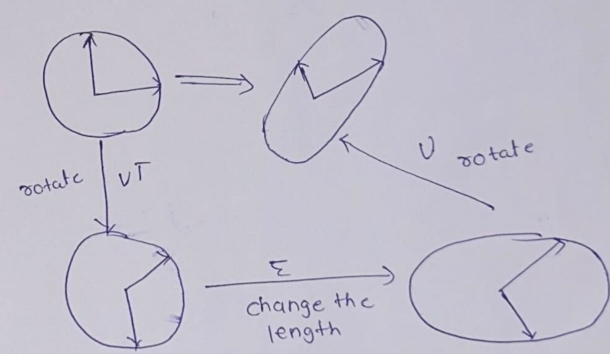
$\boxed{A^T A = V \Sigma^T \Sigma V^T}$

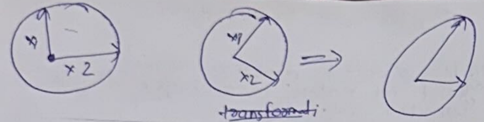
$\rightarrow$ For Finding $U$

$A A^T = (U \Sigma V^T)(V \Sigma^T U^T)$

$\boxed{A A^T = U \Sigma \Sigma^T U^T}$

Here we Rotate $U$ and $V$ and Stretch the Sigma






~~transformati~~ rotate        ellipse while stretching

for SVD
↳ ev, ev

**Example** $A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}_{2\times 3}$

$A^T A = V$

$A A^T = U$

$A^T = \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}_{3\times 2}$

For $U$: $AA^T = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}_{2\times 3} \begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}_{3\times 2} = \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix}_{2\times 2}$

→ For eigen value.

$AX = \lambda X = 0$

$(AX - \lambda X) = 0$

$(A - \lambda I)X = 0 \qquad X \neq 0$

$A - \lambda I = 0$

$\begin{bmatrix} (11-\lambda) & 1 \\ 1 & (11-\lambda) \end{bmatrix} = (11-\lambda)^2 - 1^2$

$= (11-\lambda+1)(11-\lambda-1) = 0$

$\boxed{\lambda_i = 12, 10}$ → 2 singular values

eigen vector I $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ → perform gram-schmidt orthogonal process

$\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} = U$

$A^T A = V$

$\begin{bmatrix} 3 & -1 \\ 1 & 3 \\ 1 & 1 \end{bmatrix}_{3\times 2} \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}_{2\times 3} = \begin{bmatrix} 10 & 0 & 2 \\ 0 & 10 & 4 \\ 2 & 4 & 2 \end{bmatrix}$

$m_1$

$(A - \lambda I) = 0$

$\begin{bmatrix} 10-\lambda & 0 & 2 \\ 0 & 10-\lambda & 4 \\ 2 & 4 & 2-\lambda \end{bmatrix} = \lambda^3 - S_1\lambda^2 + S_2\lambda - S_3$

$S_1$: ~~det~~ trace (A); $S_2$: minor of diagon matrix; $S_3$: det(A)

$= \lambda^3 - 22\lambda^2 + 120\lambda - 0$

$m_1 \; (20-16) = 4$

$m_2 \; (20-4) = 16$

$m_3 \; (0-20) = 100$

$\lambda^3 - 22\lambda^2 + 120\lambda = 0$

$\lambda(\lambda^2 - 22\lambda + 120) = 0$

$\boxed{\lambda_1 = 0}$ $\boxed{\lambda_2 = 12}$ $\lambda_3 = 10$

$V^T = \begin{bmatrix} v_1^T \\ v_2^T \\ v_3^T \end{bmatrix}$

$\lambda_1 = 12$

Finding V matrix $V_1$

$\begin{bmatrix} -2 & 0 & 2 \\ 0 & -2 & 4 \\ 2 & 4 & -10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 0$

By cramer's rule,

| | $x_1$ | $-x_2$ | $x_3$ |
|---|---|---|---|
| $\lambda_1$ | 4 | -8 | 4 |
| | -16 | -8 | 0 |
| $\lambda_2$ | | | |
| $\lambda_3$ | 4 | -8 | -20 |

$V = \begin{array}{c|ccc} & e_1 & e_2 & e_3 \\ \hline \text{nul} & \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 0 & -5 \end{bmatrix} \end{array}$

$$V^T = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 2 & -5 \end{bmatrix}$$ → Need this orthogonal matrix

By - Gram-schmidt process

$$V^T = \begin{bmatrix} 1/\sqrt{6} & 2/\sqrt{6} & 1/\sqrt{6} \\ 2/\sqrt{5} & 1/\sqrt{5} & 0 \\ 1/\sqrt{30} & 2/\sqrt{30} & -5/\sqrt{30} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sqrt{12} & 0 & 0 \\ 0 & \sqrt{10} & 0 \end{bmatrix}_{2\times 3}$$

$\Sigma =$ Same size of $A$

For $AA^T = U$
$\lambda = 12, 10$

For $A^TA = V$
$\lambda = 12, 10, 0$

Mainly use in Recommender System

$A^TA \rightarrow$ Symmetric that's why
$\rightarrow V =$ Orthogonal

# What is optimization in ML?

$\boxed{\text{Objective}}$ :- Minimum the difference between predicted and Actual output.

⟹ This is also called as $\boxed{\text{cost function}}$ or $\boxed{\text{loss function}}$.

## $\boxed{\text{Objective of Training Algo:}}$

⟹ Find out optimum model's params so that error can be be minimum.

Most of ML Algo does not have closed form solution to get directly optimum Model's params. that's why optimization Algo comes...

⟹ optimization in general ~~refers to~~ the task either minimizing or maximizing any function

*. What is convex function?

* Prerequisite:-

1. Hessian Matrix
2. Principal Minors

## *. How optimization help in ML ?

1. improving Model performance
   ↳ By choosing best hyperparams
2. Reducing computational cost
   ↳ when dealing with large datasets optimization technique reducing cost by finding efficient soln.
3. Avoiding overfitting
4. Handling constraints
   ↳ In real world application ML models often need to satisfy various constraint such as fairness, privacy, resouce limitations.
   ↳ optimized Technique can consider these constraint in to the model training process to ensure that resulting Model's meet the desired criteria.

## *. Application of convex / constrained convex optimization in ML ?

1. Linear Regression :- Finding best fit line by reduced the MSE or SSE.
2. logistic Regression : Aims to find the optimal params that minimize the logistic or cross entropy loss which measures diff. between predicted proba. and actual class label
3. SVM → Find the optimal hyperplane that seperates data points into different classes with maximum margin.
   ↳ where objective maximize the margin while minimizing classification error.
4. PCA → Aims to find the orthogonal Axes (principal componets) that captures maximum variance in data.
   → where objective is maximizing variance of projected data points along with the principal componets

## *. What is Hessian Matrix ?

$f : R^n \to R$ where n = number of variable

Hassian Matrix denoted as $\boxed{H_f = n \times n}$

⟹ each entry in hessian matrix denoted as represent the second order partial derivative with respect to variables.

**2 Variables**

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2} \end{bmatrix}_{2 \times 2}$$

**3 Variables**

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1} & \frac{\partial^2 f}{\partial x_1 x_2} & \frac{\partial^2 f}{\partial x_1 x_3} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2} & \frac{\partial^2 f}{\partial x_2 x_3} \\ \frac{\partial^2 f}{\partial x_3 x_1} & \frac{\partial^2 f}{\partial x_3 x_2} & \frac{\partial^2 f}{\partial x_3} \end{bmatrix}$$

## *. What is principal Minors ?

⟹ Submatrix of A is the $k*k$ which are obtain from $n*n$ by deleting $\boxed{n-k \text{ cols and rows}}$ where $k$ is the order of principal minor.

① First order principal Minor

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{array}{l} K = 1 \\ n = 3 \\ \boxed{n-K = 2} \end{array}$$

Principal Minor = $a_{11}$

Normal Minor = $a_{22}$, $a_{33}$

② second order principal Minor

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \begin{array}{l} K = 2 \\ n = 3 \\ n-K = 1 \end{array}$$

Principal Minor = $\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$

Normal minor = $\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$

## *. Convex Optimization

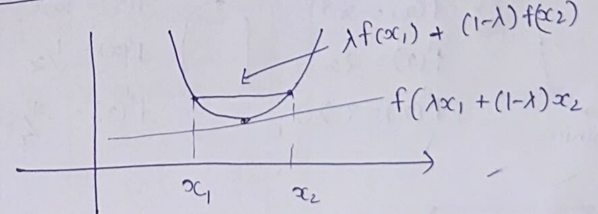
**Prerequisite :-**

① what is convex and concave function

## * what is convex/concave fun.?

⟹ A function $f(x)$ is convex if for all $x_1$ and $x_2$ in domain of X and for all $\lambda$ in the interval $[0, 1]$ the following equality holds.

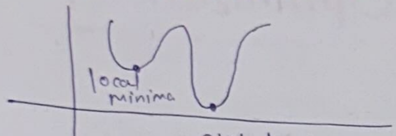
$$\boxed{f(\lambda x_1 + (1-\lambda)x_2) \le \lambda f(x_1) + (1-\lambda) f(x_2)}$$




⟹ If pick any 2 point and line segments joining them and never cross the curve than these type function called convex function.

$\boxed{\text{local minima} = \text{Global minima}}$

## * Non convex func.

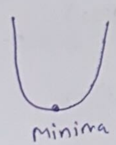



$\boxed{\text{local minima} \neq \text{global minima}}$

## *. How to check function is convex or not ?

### For 2D function

If $\boxed{f'' \geq 0}$ for all domain points than function is convex.



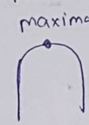

Ex $f(x) = x^2$

$f'(x) = 2x$

$f''(x) = 2 > 0$

**Convex**

If $\boxed{f'' \leq 0}$ for all domain points than function is concave.




$f(x) = \log x, \sqrt{x}$

$f'(x) = 1/x$

$f''(x) = -1/x^2 < 0$

**Concave**

If $f''(x) \leq 0$ and $f''(x) \geq 0$ for some points of domain than it is neither convex nor concave

Ex $f(x) = x^3$ $f'(x) = 3x^2$ $f''(x) = 6x$

$x > 0 \rightarrow$ convex

## For Higher Dimension :-

$\Rightarrow$ There are 2 methods using principal minors and eigen Values

| f(x) is convex | f(x) is strictly convex |
|---|---|
| H is $\boxed{\text{positive semi definite}}$ | H is $\boxed{\text{positive definite}}$ |
| where $D_1 > 0,\ D_2 \geq 0,\ D_3 \geq 0$ | where $D_1 > 0,\ D_2 > 0,\ D_3 > 0$ |
| At least one of the $D_i = 0$ where $i = 1 \ldots n$ | |
| $\boxed{\text{* D = principal minors det}}$ | |

f(x) is concave

H is $\boxed{\text{Negitive semi definite}}$

where

$D_1 < 0,\ D_2 > 0,\ D_3 < 0$

Alternative sign but First principal minor should be Negative

**Ex 01** $f(x, y) = x^2 + y^2$

→ Second order partial derivative

$$\frac{\partial^2 f}{\partial x^2} = 2 \qquad \frac{\partial^2 f}{\partial y^2} = 2 \qquad \frac{\partial^2 f}{\partial x \partial y} = 0$$

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Prinicipal Minor

$D_1 = 2 > 0$

$D_2 = \begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} = 4 > 0$

{ For diagonal Matrix diagonal entries are the eigen Value }

→ Both the minor are $> 0$ than function is $\boxed{\text{convex}}$

**Ex-02** $f(x, y) = -x^2 - y^2$

$$\frac{\partial^2 f}{\partial x^2} = -2 \qquad \frac{\partial^2 f}{\partial y^2} = -2 \qquad \frac{\partial^2 f}{\partial x \partial y} = 0$$

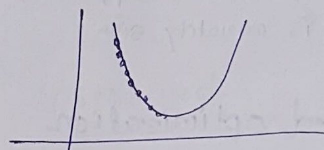
$$H = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} \qquad D_1 = -2 \qquad D_2 = 4 > 0$$

Here is Alternative sign and First minor is Negative than function is $\boxed{\text{Concave}}$

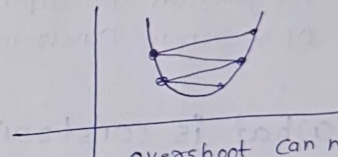
⟹ For convex optimization Mainly Gradient descent based can considered.

⟹ In this optimization technique criteria is $\boxed{\text{Objective function is differentiable}}$.

⟹ In this Algo MIMP param is $\boxed{\text{step size}}$



Take More time to converge



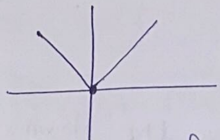
Overshoot can not find optimal soln.

## *. IMP points for convex func.

⟹ A convex ~~concave~~ function not need to be differentiable.

⟹ A convex function nod need to be continous.

at $|x| = 0$ not differentiable



⟹ A set of convex function's sum is always be convex function.

## *. Application of convex optimization

① linear Regression

② logistic Regression

# * Constoained convex optimization

## * Pre requisite :-

⟹ langoagian Moltiplier

⟹ langoagian Moltiplier method is applicable only when constraint is equality sign.

## * What is constoained optimization problem ?

⟹ A general non linear programming problem with equality constraint

$$\text{maximize / minimize} \quad Z = f(x)$$

$$\text{s.t.} \quad g(x) = 0$$

$$x \geq 0$$

→ Assume that $f(x)$ and $g(x)$ differentiable wrt $x$.

## * Neccessary Condition

⟹ To find Neccessary condition for the Maxima or Minima value of $Z$; a new function is formed by introducing a multiplier $\lambda$ known as langrage Multiplier

$$L(x, \lambda) = f(x) + \lambda g(x)$$

where $\lambda$ is constant unrestricted in sign

## * purpose of langrangian Function :

| Maximize / minimize<br>$Z = f(x)$<br>s.t. $g(x) = 0$<br>$x \geq 0$<br>constrained optimization | maximize / minimize<br>$L(x, \lambda) = f(x) + \lambda g(x)$<br>s.t. $x \geq 0$<br>Unconstoained optimization |
|---|---|

## * Neccessary Condition :

$$\frac{\partial L}{\partial x} = 0, \quad \frac{\partial L}{\partial \lambda} = 0$$

After we get stationary points $(x^*, \lambda^*)$

## * Sufficient condition :-

### First Method :

⟹ constraints are equality sign than neccessary condition becomes sufficient conditions for Maxima / Minima

⟹ For Minima you need to prove function is convex.

## Second Method

⟹ After converting unconstrained optimization problem

⟹ We need to define Boardered Hessian matrix $H^B$.

$$H^B = \begin{bmatrix} 0 & U \\ U^T & V \end{bmatrix}_{(m+n)\times(m+n)}$$

where m = No. of constrai
n = No. of Variables

$$U = \left[\frac{\partial g}{\partial x}\right] \qquad V = \left[\frac{\partial^2 L}{\partial X}\right] \leftarrow \text{Hessian matrix}$$

⟹ At the stationary point.

starting from $(2m+1)$ principal minor and compute upto total number of $(n-m)$ pricipal minor of the $H^B$ at point $(x^*, \lambda^*)$ is

⟹ **Maximum** : principal minor form an alternative sign with $(-1)^{m+n}$

⟹ **Minimum** : principal minor sign of $(-1)^m$.

* **Constraint is equality sign than use simple Hassian matrix :-**

→ Find Necessary condition for stationary points

→ Find Hessian Matrix

→ Find principal Minors

→ check function is minima or maxim

based on that stationary points are Minimum or Maximum

Ex 01) Minimize

$$Z = 2x_1^2 - 24x_1 + 2x_2^2 - 8x_2 + 2x_3^2 - 12x_3 + 200$$

S.t : $x_1 + x_2 + x_3 = 11$

$x_1, x_2, x_3 \geq 0$

⟹ langragian function.

$$L(x,\lambda) = \begin{pmatrix} 2x_1^2 - 24x_1 + 2x_2^2 - 8x_2 \\ + 2x_3^2 - 12x_3 + 200 \end{pmatrix} + \lambda(x_1 + x_2 + x_3 - 11)$$

⟹ Neccessary condition for Find stationary point

$$\frac{\partial L}{\partial x_1} = 0 \Rightarrow 4x_1 - 24 + \lambda = 0 \Rightarrow x_1 = \frac{-\lambda + 24}{4}$$

$$\frac{\partial L}{\partial x_2} = 0 \Rightarrow 4x_2 - 8 + \lambda = 0 \Rightarrow x_2 = \frac{-\lambda + 8}{4}$$

$$\frac{\partial L}{\partial x_3} = 0 \Rightarrow 4x_2 - 12 + \lambda = 0 \Rightarrow x_3 = \frac{-\lambda + 12}{4}$$

$$\frac{\partial L}{\partial \lambda} = 0 \Rightarrow x_1 + x_2 + x_3 - 11 = 0$$

$\hookrightarrow \boxed{\lambda = 0}$

Stationary point $(X; \lambda) = (6, 2, 3; 0)$

$$\frac{\partial^2 L}{\partial x_1^2} = 4 \qquad H = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\frac{\partial^2 L}{\partial x_2^2} = 4$$

$$\frac{\partial^2 L}{\partial x_3^2} = 4$$

$D_1 = 4 > 0$

$D_2 = 16 > 0$ $\quad$ K, 2 $\quad$ 3 - 2

$D_3 = 64 > 0$

$\longrightarrow$ All minors positive therefore it is $\boxed{\text{Positive definite}}$ therefor it is $\boxed{\text{convex function}}$

$\longrightarrow$ All stationary point is Minimum

# KKT Condition

$\Rightarrow$ KKT condition based first derivative tests (Sometimes called first order necessary conditions)

$\Rightarrow$ KKT condition used when inequality constraints available

| | | |
|---|---|---|
| maximize/minimize $f(x)$ st. $g(x) = b_i$ | $\Rightarrow$ | Solve problem with langrangian Multiplier |
| maximize / minimize $f(x)$ st. $g_i(x) \le b_i$ or $g_i(x) \ge b_i$ | $\Rightarrow$ | Solve with KKT condition |

* Necessary and sufficient cond.

## ① Maximization problem :-

Ex maximize $f(x)$

s.t. $g_i(x) \le b_i$

⟹ convert each $i^{th}$ inequality constraints into equation by adding non negative slack variable $s_i^2$.

$$\left\{ \begin{array}{l} x_1 + x_2 \le 1 \\ x_1 + x_2 + s_i = 1 \end{array} \right.$$

→ slack variable

⟹ Here problem is Non linear that's why we added slack variable $s_i^2$.

$g_i(x) \le b_i \Rightarrow g_i(x) + s_i^2 = b_i$

$$\hookrightarrow \boxed{h_i(x) = g_i(x) + s_i^2 - b_i = 0}$$

maximize $f(x)$

s.t $h_i(x) = 0$

⟶ Now Apply the langrangian multiplier

$$L(x, s, \lambda) = f(x) - \sum_i \underbrace{\lambda_i}_{\text{langrangian multiplier}} h_i(x)$$

$$L = f(x) - \sum_i \lambda_i \left( g_i(x) + s_i^2 - b_i \right)$$

* Necessary condition for langrangian multiplier for stationary point

$$\frac{\partial L}{\partial x} = 0 \Rightarrow \frac{\partial f}{\partial x} - \sum \lambda_i \frac{\partial g_i}{\partial x} = 0 \quad — ①$$

$$\frac{\partial L}{\partial \lambda_i} = 0 \Rightarrow g_i(x) + s_i^2 - b_i = 0 \quad — ②$$

$$\frac{\partial L}{\partial s_i} = 0 \Rightarrow -2\lambda_i s_i = 0 \quad — ③$$

⟶ After solving this we get stationary point

⟶ This stationary we check either it is minima or maxima.

⟶ Multiply eq ③ with $\frac{1}{2} s_i$

$$\boxed{\lambda_i s_i^2 = 0}$$

from eq ② $\boxed{s_i^2 = b_i - g_i(x)}$

$$\boxed{\lambda_i \left( b_i - g_i(x) \right) = 0} \quad — ④$$

| | |
|---|---|
| $\lambda_i = 0$<br>→ $\lambda_i$ measure the rate of variation of f with respect to the $b_i$ | $b_i = g_i(x)$ constrained are satisfy as equality |

→ From eq (3)

| $\lambda_i = 0$, $s_i \neq 0$ | $s_i = 0$, $\lambda_i \neq 0$ | both are 0 |
|---|---|---|
| → constraint are satisfy as inequality <br> $\boxed{g_i(x) + s_i^2 = b_i}$ <br> ⇒ so we can relaxed constraint It will not affected stationary point | → constrained are satisfy as equality <br> $\boxed{g_i(x) = b_i}$ <br> $\lambda_i < 0$ <br> ⇓ <br> $\frac{\partial f}{\partial b} < 0$ <br> $b_i \uparrow f(x) \downarrow$ <br> for Maximization this case not Applicable. | |

Hence for Maximization $\lambda_i \geq 0$

minimization $\lambda_i \leq 0$

$= 2x^2 - 24x_1 + 2x_2^2 - 8x_2 + 2x_3^2 - 12x_3$

## Final Conclusion

| Maximize $f(x)$ s.t $g_i(x) \leq b_i$ | Minimize $f(x)$ s.t $g_i(x) \leq b_i$ |
|---|---|
| $\frac{\partial f}{\partial x} - \sum \lambda_i \frac{\partial g_i}{\partial x} = 0$ | $\frac{\partial f}{\partial x} - \sum \lambda_i \frac{\partial g_i}{\partial x} = 0$ |
| $\lambda_i (g_i(x) - b_i) = 0$ | $\lambda_i (g_i(x) - b_i) = 0$ |
| $g_i(x) \leq b_i$ | $g_i(x) \leq b_i$ |
| $\lambda_i \geq 0$ | $\lambda_i \leq 0$ |

Ex 01 **Minimization problem**

(1) Min $Z = (x_1 - 2)^2 + (x_2 - 1)^2$

s.t $x_1^2 - x_2 \leq 0$, $x_1 + x_2 \leq 2$

and $x_1, x_2 \geq 0$

→ define the langrangian function L as

$L = f - \lambda_1 g_1 - \lambda_2 g_2$

$L = (x_1-2)^2 + (x_2-x_1)^2$
$- \lambda_1 (x_1^2 - x_2) - \lambda_2(x_1+x_2-2)$

For Minimization Neccesary contionL

$\frac{\partial L}{\partial x} = 0 \rightarrow \frac{\partial L}{\partial x_1} = 2(x_1-2) - 2\lambda_1 x_1 - \lambda_2 = 0$

$\frac{\partial L}{\partial x_2} = 2(x_2-1) + \lambda_1 - \lambda_2 = 0$

only calculate this

$\lambda_i g_i = 0 \rightarrow \lambda_i(x_i^2 - x_2) = 0$

$\lambda_2(x_1+x_2-2) = 0$

$\boxed{\lambda_i \leq 0}$ → $\lambda_1, \lambda_2 \leq 0$

$g_i \leq 0$
$x \geq 0$
feasible region

→ $x_i^2 - x_2 \leq 0$

$x_1 + x_2 \leq 2$

→ $x_1, x_2 \geq 0$

For Maximization problem

$\boxed{\text{KKT condition} = \text{Sufficient condition}}$

$z = 2x_1^2 - 24x_1 + 2x_2^2 - 8x_2 + 2x_3^2 - 12x_3$

| Maximization | Minimization |
|---|---|
| Case-01<br>Maxima $f(x)$ concave<br>S.t $g_i(x) \leq b_i$ convex | case-03<br>minimize $f(x)$ convex<br>S.t. $g_i(x) \leq b_i$ convex |
| Case-02<br>maxima $f(x)$ concave<br>S.t $g_i(x) \geq b_i$ concave | Case-04<br>minimize $f(x)$ Convex<br>S.t $g_i(x) \geq b_i$ concave |

→ use the Hessian Matrix

→ use the eigen values

## Example

minimize $Z = -\log x_1 - \log x_2$

S.t = $x_1 + x_2 \leq 2$

$x_1, x_2 \geq 0$

minimize
→ convex
$\leq$ → convex

→ here we need prove both function is convex.

⇒ For $f(x)$ to be convex

$$H_{f(x)} = \begin{bmatrix} 1/x_1^2 & 0 \\ 0 & 1/x_2^2 \end{bmatrix}$$

$d_1 \geq 0$
$d_2 > 0$

pdefiti
↓
$\boxed{\text{Convex}}$

$\Rightarrow$ For $g(x)$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \begin{matrix} d_1 \geq 0 \\ d_2 \geq 0 \end{matrix} \longrightarrow PSD \rightarrow \text{convex}$$

$\Rightarrow$ From this KKT condition sufficiat for minimu

$$L = (-\log x_1 - \log x_2) - \lambda(x_1 + x_2 - 2)$$

**Nececcary condition**

searching sp.

$$\frac{\partial L}{\partial x} = 0 \; ; \quad \lambda g = 0 \; ; \quad \lambda \leq 0 \; ; \quad g \leq 0 \; ; \quad x \geq 0$$

$-1/x_1 - \lambda = 0$ — ①

$-1/x_2 - \lambda = 0$ — ②

$\lambda(x_1 + x_2 - 2) = 0$ — ③

$\lambda \leq 0$ — ④

$x_1 + x_2 \leq 2$ — ⑤

$x_1, x_2 \geq 0$ — ⑥

case = 01 $\quad \lambda = 0$

$-1/x_1 = 0$

$x_1 = -\infty$ or $\infty$

Not satisfy ⑥ $\quad$ Not satisfy ⑤

case = 02 $\quad \lambda \neq 0$

$x_1 + x_2 - 2 = 0$

$x_1 = -1/\lambda \qquad x_2 = -1/\lambda$

$\boxed{\lambda = -1} \rightarrow$ satisfy = 3

$\boxed{x_1 = 1, \; x_2 = 1}$

satisfy ⑤ and ⑥

Stationary point

$(x_1, x_2; \lambda) = (1, 1, -1)$