

★ Linear Transforms

$T: X \rightarrow Y$ where $X \in \mathbb{R}^n, Y \in \mathbb{R}^m$

& $T \rightarrow$ Transformation matrix $(m \times n)$

$m > n \Rightarrow T$: Tall matrix.

$m < n \Rightarrow T$: Fat matrix.

Corresponding to any linear transform there exists a transform matrix.

e.g. Scaling, Rotation, Translation, Reflection, etc.

$Ax = b \Rightarrow x \rightarrow$ input vector

$b \rightarrow$ output vector

& $A \rightarrow$ Transformation Matrix

Fundamental subspaces of vectors :-

1) Column space

3) Null space

2) Row space.

4) Left Null space.

(Column space of transpose)

(Null space of $(\text{matrix})^T$)

$$\dim(C(A)) = \dim(C(CAT)) = \text{rank}(A)$$

* dim → dimensionality :- No. of linearly independent vectors.

* Rank-Nullity Theorem :-

If Number of linearly independent rows are less than the dimensionality of matrix then there will be at least one element in the nullspace & hence matrix is not invertible.

i.e. if $NCA) \neq \phi \Rightarrow |A| = 0$

18/2/2021

* MATRICES

To check if $Ax=b$ has a solution

i) check $\text{rank}(A) = \text{rank}(Ab)$

OR ii) check if $|A| \neq 0$. i.e. A is a full rank matrix

inverse Operation

$\xrightarrow{\quad}$ well posed system
 $\xrightarrow{\quad}$ ill posed system

If small change in data gives large change in solution then we say that it is **ill posed**

$Ax = b$ where $A \rightarrow$ operator
 $x \rightarrow$ original image
 $b \rightarrow$ blurred image

then inverse operation is getting original image from blurred image.

* if this is possible then it is called well posed system.

For well posed system we have

- ① uniqueness ② existence
- ③ continuous dependance.

* condition-ness of system

close to 1 \Rightarrow well conditioned system

if $C_N \gg 1 \Rightarrow$ ill conditioned system

\hookrightarrow condition number.

One way of defining C_N is

$$C_N = \frac{\text{max. eigen value}}{\text{min. eigen value.}}$$

if min eigen value = 0 \Rightarrow ill conditioned

Given $Ax=b$ & if b -belongs to $C(A)$
then we say that system is solvable

Diagonal Dominance :-

Given $A = [a_{ij}]_{n \times n}$

if $|a_{ii}| \geq \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}| \Rightarrow A$ is diagonally dominant

Gauss elimination :-

$A = LU$, where A is square & invertible.

prod. (diag(U)) = $|A|$ ← IMP

* LU Algorithm :-

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

$$U = A$$

```
for i=1 to n-1  
  for j=i+1 to n
```

$$L(j,i) = U(j,i)/U(i,i)$$

$$U(j,:) = U(j,:)-L(j,i) \times U(i,:)$$

QR Decomposition:-

$$A = QR$$

- i) A can be non-square
- ii) columns of A are linearly indep.
- iii) Q \Rightarrow orthonormal matrix
- iv) R \Rightarrow upper triangular matrix
- v) Q will always have left inverse.
- vi) If A is square, Q will also have right inverse.

✓ $\Rightarrow Q^T Q = I$

vi) $\Rightarrow Q Q^T = I$

& $Q^T A = R$

If A is square then Q is called unitary matrix.

i.e. $Q^T Q = Q Q^T = I$

Normed Linear spaces :-

Norm \Rightarrow length of a vector.

i.e. $\|x\|_p = (x_1^p + x_2^p + x_3^p + \dots + x_n^p)^{1/p}$

Properties of Norm :-

- ① norm is always +ve.
- ② $\|x+y\| \leq \|x\| + \|y\|$

* Inner product:

$$\langle x \cdot x \rangle = x^T x = x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2$$

i.e. inner product of a vector gives squared L2 norm.

* Space in which norm is induced by inner product is called Hilbert space.

* orthornormality \Rightarrow

$$\langle q_i \cdot q_j \rangle = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

norm of orthonormal vector = 1

* For any orthogonal matrix, left inverse always exists.

i.e.

$$Q^T Q = I$$

And if, (in $A = QR$), A is square & all the columns are orthonormal then right inverse also exists. & Q is called unitary.

i.e.

$$Q^T Q = Q Q^T = I$$

Properties of unitary transformation:

① Unitary transformations preserves length of vector.

i.e. given a , Q , unitary matrix

$$\& \quad Qx = y \text{ then } \|x\| = \|y\|$$

② also unitary matrix performs -
compaction on vector.

i.e. $y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$ & $y_1 > y_2 > y_3 \Rightarrow$ compaction

* norm is also implied as energy
& we say that energy is compacted.

③ Data uncorrelation \Rightarrow in unitary transformation , correlated data becomes uncorrelated .

Gram-Schmidt Orthogonalization:

Given a set of linearly independent vectors, G-S converts those into orthonormal vectors.

$$A = \begin{bmatrix} | & | & | \\ u_1 & u_2 & \dots & u_n \\ | & | & | \end{bmatrix}, Q = \begin{bmatrix} | & | & | \\ q_1 & q_2 & \dots & q_n \\ | & | & | \end{bmatrix}$$

then using GS orthogonalization we have,

$$v_1 = u_1 \quad \& \quad q_1 = v_1 / \|v_1\|$$

$$v_2 = u_2 - \frac{u_2^T v_1}{v_1^T v_1} v_1 \quad \& \quad q_2 = \frac{v_2}{\|v_2\|}$$

$\|y_i\|$

$$v_n = u_n - \left[\sum_{i=1}^{n-1} \frac{u_n^T v_i}{v_i^T v_i} v_i \right]$$

Eigen Decomposition:-

Any sq. mat A can be decomposed into

$$A = X \Sigma X^{-1}$$

$\Sigma \rightarrow$ diagonal matrix of eigen values.

$X \rightarrow$ matrix with columns=eigen vectors

* Eigen vectors gets scaled, not rotated, during transformation

$$\therefore A^T = (X^{-1})^T \Sigma^T X^T$$

if A is

symmetric,

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

$$\Rightarrow$$

$$X^{-1} = X^T$$

$$\therefore A A^T = A^T A = I$$

$\therefore A$ is unitary.

• For symm. matrix eigen vectors are orthonormal & eigen values are real.

e.g. $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \rightarrow$ symmetric \rightarrow yields real roots.

$A = \begin{bmatrix} 1 & 1 \\ -1 & 2 \end{bmatrix} \rightarrow$ Antisymmetric \rightarrow yields complex / imaginary roots.

\therefore For symmetric A , we have

$$X^{-1} = X^T$$

$$A = X \Sigma X^T$$

Note :-

For finding eigen decomposition it is not necessary to write program,
In built function can be used.

* No. of Non-zero eigen values denotes rank of matrix.

of

* No. of zero eigen values ^ matrix denotes dimensions of null space of that matrix.

* positive Definite matrix :-

$\langle z, Az \rangle$ i.e. dot product of z & Az .

① if $\overrightarrow{x^T A x} \geq 0$ & $\bar{z} \neq 0$, then
A is called, positive semi - de
finite. (i.e. $\forall_i \underline{\lambda_i \geq 0}$)

② if $x^T A x > 0, \Rightarrow A$ is positive
definite.

i.e. $\forall_i, \underline{\lambda_i > 0}$

SPD matrices :-

(symmetric positive definite)

a spd matrix A can be decomposed into

$$A = X \Sigma X^T \quad \& \quad \Sigma = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

S P S D \rightarrow symmetric positive semi-definite

then if A is S P S D,
we have,

$$A^* = X^* \Sigma^* X^{*\top}$$

is done by removing all zero eigen values.

By doing this, we avoid ill-conditionedness & This is called preconditioning the matrix. This avoids ill-conditioned system.

- * pre conditioning a matrix \Rightarrow Removing zero eigen values.
- * 2-similar matrices have same eigen values.
- * Matrices A & B are similar iff for some matrix M ,

$$A = M^{-1}BM$$

- * AB & BA are similar & $M = \underline{\underline{B}}$
 - i.e. $AB = \tilde{B}^{-1}(BA)B$
- Also $A^T A$ & $A A^T$ are similar & \therefore has same eigen values.

* Similar Matrices has same eigen values.

* Singular Value Decomposition:

Any matrix A can be decomposed into U, Σ, V^T such that,

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

where, U & V are unitary matrices & $\Sigma \rightarrow$ diagonal matrices.

of $\sigma \rightarrow$ singular values $\doteq \sqrt{\lambda}$ eigenvalues

- Unitary matrix \rightarrow performs rotation preserves length

- Diagonal matrix \rightarrow performs scaling preserves orientation

$\therefore SVD \Rightarrow$ Rotation-scaling-rotation

If we are given a matrix A which is rectangular. then $A^T A$ is always square.

$$\text{then } \underset{n \times p}{\text{SVD}(A)} = \underset{n \times n}{U} \underset{n \times p}{\Sigma} \underset{p \times p}{V^T}$$

where, U & V are unitary & orthogonal & columns of U = left singular vector.
i.e. columns of U = eigen vectors of $A^T A$

& $\Sigma \Rightarrow$ diagonal mat of singular values arranged in descending order. & singular values are square roots of eigen values of $A^T A$ or $A A^T$ & singular values are always real.
& rows of V^T = right singular vectors
i.e. rows of V^T = eigen vectors of $A A^T$

$$\text{Now, } A = U \Sigma V^T \xrightarrow{\text{SVD}}$$

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

as Σ is diagonal, $\Rightarrow \Sigma = \Sigma^T$

$$\therefore A A^T = U \Sigma V^T V \Sigma U^T$$

$$\therefore A A^T = U \Sigma^2 U^T$$

As V is unitary
 $\therefore V^T V = I$

Truncated SVD:-

if we discard all 6's i.e. eigen values that are zero, then

$$\tilde{A} = \sum_{i=1}^k \sigma_i v_i v_i^T$$

where \tilde{A} is well conditioned & it is an approximation of A

Also, if U_K & V_K are truncated matrices then.

$$A = U_K U_K^T A$$

$$\& \quad A = A V_K V_K^T$$

* SVD for reducing dimensions of vector :-

* principle component analysis to know which data properties are useful & preserving those

$$\frac{(A - M)(A - M)^T}{N} = \text{covariance matrix.}$$

$M \rightarrow$ mean matrix

$N \rightarrow$ no. of datapoints / sets.

$$\text{if } A' = \frac{(A - M)(A - M)^T}{N} = U_{n \times n} \Sigma_{n \times m} V_{m \times m}^T$$

↑
principle component matrix.

e.g. if $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

then $M = \text{column mean} = 2, 3$

$$\therefore A - M = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\therefore C = (A - M)(A - M)^T = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \times \frac{1}{N}$$

$$\therefore \text{covariance matrix } C = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \times \frac{1}{n}$$

we can neglect $\frac{1}{n}$, because it does not affect maximization

$$\therefore C = (A - M)(A - M)^T$$

eigen vectors of $C \Rightarrow$ principle component

$$SVD(A - M) = \bigcup_{n \times n} S V^T$$

where, $S = \text{diag}(6, 6_2, 6_3, \dots)$

$\bigcup_{n \times n}$ eigen vectors of $C = \frac{\text{principle}}{\text{component}}$

Note in program $\text{SVD}(A - M) \Rightarrow \text{SVD}(A - M)(A - M)^T$

we want to maximize $x^T C x$
where

x - turns out to be eigen
vectors of C .

Also eigen vector \rightarrow corresponding to
first eigen value \Rightarrow first principle
component, which is along highest
variance in data, and so on.

* Principle component are ~~for to~~ each other.

* principle component are ~~perp~~ to one another.

* Preservation of data variation:-

Principle component analysis preserves variation in data.

$\hat{x} = UU^T x \rightarrow$ also $\hat{x} = x \sqrt{V^T}$

$\hat{x} = UU^T x$ → reconstruction of data.

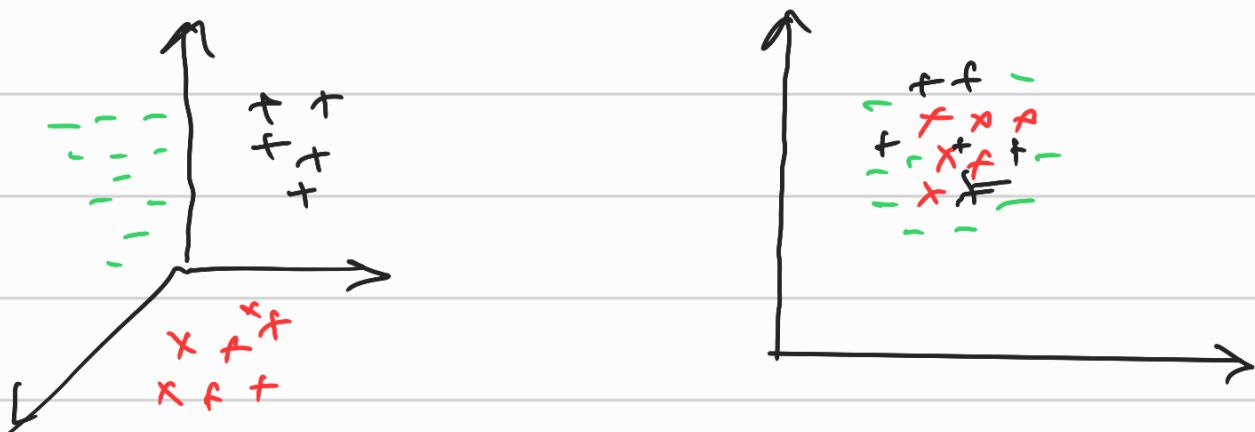
where \hat{x} is projection of x from higher dimension to lower dim.

- So, if we have k -principle components then they represent basis for k -dimensional space.
- Also, Any ' n '-dimensional vector can be projected on k -dimensional space of principle component without loosing data.

i.e. $x \rightarrow n$ -dimesional vector, then

$U_k^T x_n = y_k \Rightarrow y_k = \text{projection of } x_n$
from $n \rightarrow k$ dimensional space.

* Principle component analysis is not suitable for classification. Because even though data is separated in higher dimension, it gets clustered in lower dimension.



Higher dimension \longrightarrow lower dimension .

* PCA is mainly used for dimension reduction.

* Kernel - PCA

Used to convert data from lower dimension (\mathbf{x}) to higher dimension space (\mathbf{X})

i.e. $x \xrightarrow{\phi(x)} X$

where $\phi(x)$ = kernel function

Also, Kernel function $K(x, y)$ is given by

$$\boxed{\phi(x)^T \phi(y) = K(x, y)}$$

e.g. of kernel functions,

① $\langle x, y \rangle^n$ ② $e^{-\|x-y\|^2/2} \Rightarrow RBF$

(Radial basis kernel function)

Matrix corresponding to $K(x, y)$ is symmetric positive definite.

$U^T A$ is projection of A on principal component &

as $A = U \Sigma V^T$

Hence $U^T A = \Sigma V^T$

Σ = square of eigen values of A

V^T = eigen vectors of A .

* Reconstruction of Data in kernel PCA is not possible, that's why we always project.

* Out of sample projections :-

if we have x_1, x_2, \dots, x_n in dataset & we want to project x_0 , then it is called out-of-sample projection & It can be found as follows

$$\text{out of sample projection} = \underbrace{\Sigma^{-1} V^T A^T x_0}_{U^T}$$

* Out of sample reconstruction :-

as, $\hat{x} = U \underbrace{U^T x_0}_{\downarrow \rightarrow}$

$$\therefore \hat{x} = (A V \Sigma^{-1}) (\Sigma^{-1} V^T A^T x_0)$$

But we cannot compute \hat{x} because

we cannot compute A.

Therefore,

- ① we can compute, projection & out of sample projection of data.
- ② we cannot compute reconstruction & out of sample reconstruction of data.

* Centralizing kernel Matrix (K) :-

* $[I - \frac{1}{N} (I I^T)] K$ ⇒ This subtracts column mean from respective columns. ⇒ centralizing in terms of columns

* $K \times [I - \frac{1}{N} (I I^T)]$ ⇒ This subtracts row mean from corr. rows. ⇒ centralizing in terms of rows where $I \rightarrow$ column vector with no. of rows = no. of rows in K.

&
 $N \Rightarrow$ No. of samples / No. of columns or rows of K.

*

$$I - \frac{1}{N} (1 1^T) = \text{centralizing matrix}$$

* PCA as optimization problem :-

we want to find x which maximizes
 $\|A^T x\|$

maximizes $x, (A^T x)^T (A^T x)$

subject to, $x^T x = 1$.

maximize, $x, x^T S x,$

subject to $x^T x = 1$

This can be written in Lagrangian form,

$$L(x, \lambda) = \max_x \left\{ x^T S x - \lambda (x^T x - 1) \right\}$$

$\lambda \rightarrow$ Lagrangian parameter.

$$\text{Now, } \frac{dL}{dx} = 0 \Rightarrow \frac{d}{dx} [x^T s_x - \lambda x^T e] = 0$$

Note:- go through vector differentiation

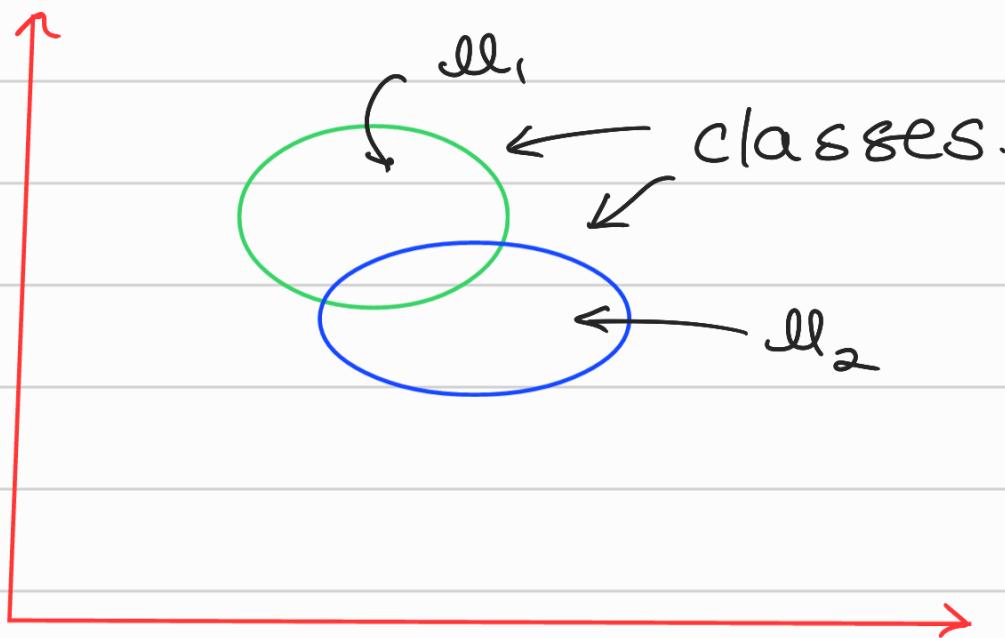
* Linear Discriminant Analysis or Fischer's LDA

- ① Mostly used for classification of Data.
- ② In LDA no. of vector depends on no. of classes.
- ③ LDA's are also eigen vectors but not same as PCA.
- ④ In LDA we find vector for classification of data & in PCA we find vector for dimension reduction

① within class variance

② between class variance.

we want to minimize 1st & maximize second. so the data is neatly separated.



Thus we are looking for $n-1$ discriminant if there are n classes. (i.e. ^{we're} looking for $n-2$ LD vectors)

Also in FLDA, we want to find a vector ω , which minimizes within class variance and maximizes betwⁿ class variance.

$$\underset{\omega}{\text{maximize}}, \quad \left\| (\mathbf{el}_1 - \mathbf{el}_2)^T \omega \right\|_2^2$$

$$\text{i.e. } \max (\mathbf{el}_1 - \mathbf{el}_2)^T \omega$$

$$\therefore \max (\omega^T (\mathbf{el}_1 - \mathbf{el}_2) (\mathbf{el}_1 - \mathbf{el}_2)^T \omega)$$

$$\text{where, } S_B = (\mathbf{el}_1 - \mathbf{el}_2) (\mathbf{el}_1 - \mathbf{el}_2)^T$$

= between class scatter matrix.

$$\& \quad S_\omega = \text{within class sc. matrix.}$$

$$\text{i.e. } \max \omega^T S_B \omega$$

$$\& \min \omega^T S_\omega \omega$$

$$\text{where, } S_\omega = \Sigma_1 + \Sigma_2$$

$$\& \Sigma_1 = (\mathbf{x}_1 - \mathbf{m}_1) (\mathbf{x}_1 - \mathbf{m}_1)^T \rightarrow \text{for class 1}$$

$$\Sigma_2 = (\mathbf{x}_2 - \mathbf{m}_2) (\mathbf{x}_2 - \mathbf{m}_2)^T \rightarrow \text{for class 2}$$

In turn,

$$\max_{\omega},$$

$$\frac{\omega^T S_B \omega}{\omega^T S_\omega \omega}$$

\rightarrow Rayleigh's method

or $L(\omega, \lambda) = \max (\omega^T S_B \omega - \lambda \omega^T S_\omega \omega)$
↪ Lagrange's method.

$$\therefore \frac{\partial L}{\partial \omega} = 0$$

$$\Rightarrow S_B \omega - \lambda S_\omega \omega = 0$$

$$\therefore S_B \omega = \lambda S_\omega \omega$$

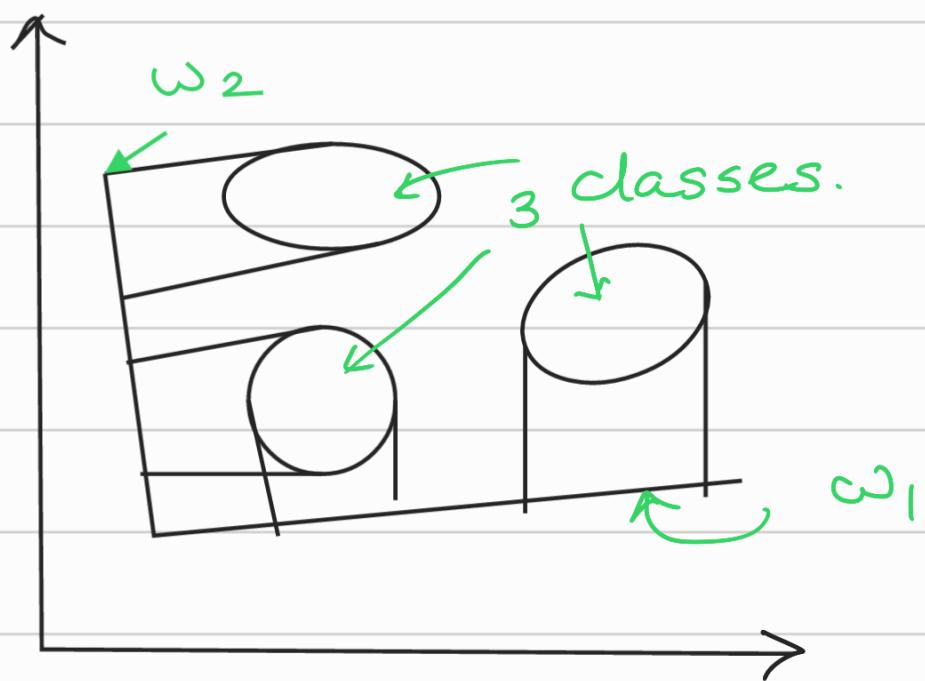
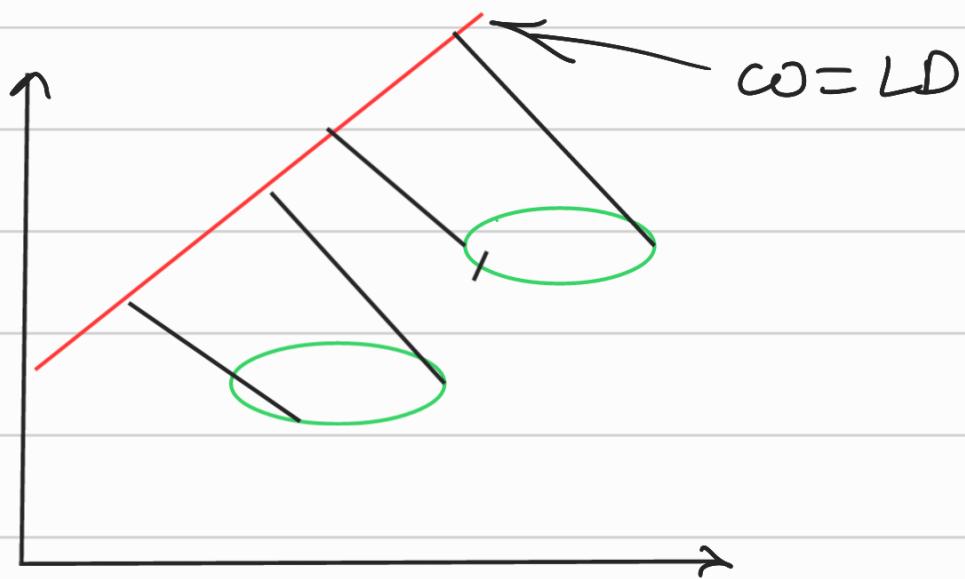
$$\therefore (S_\omega^{-1} S_B) \omega = \lambda \omega$$

which is similar to $Ax = \lambda x$

$\therefore \omega \rightarrow$ eigen vectors of $(S_\omega^{-1} S_B)$

S_ω should be a full rank matrix,
else we will have to find approximate
inverse (i.e. using SVD)

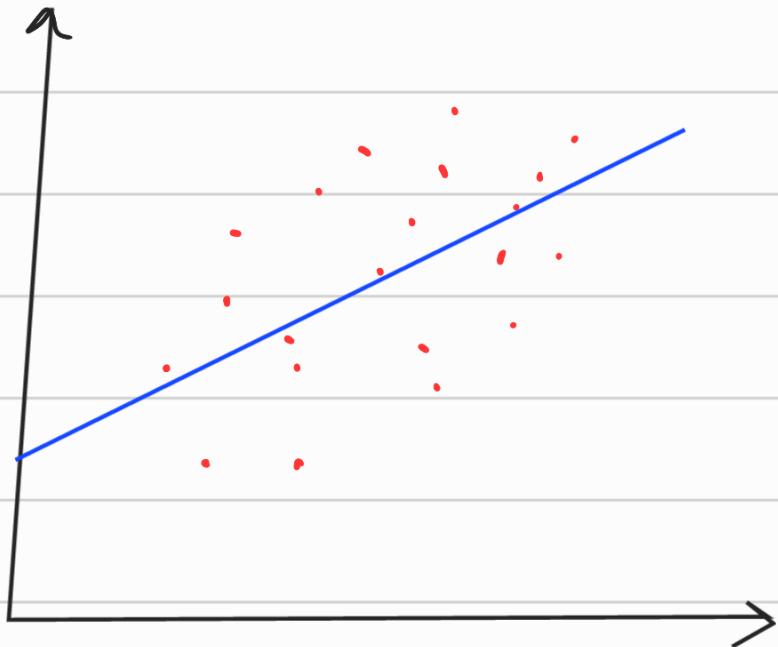
So, when rank of $(S_\omega^{-1} S_B)$ is 1,
then we get exactly 1 $\rightarrow \omega \rightarrow$ which is
called LD & This process is called
FLDA.



\therefore for n classes we need $n-1$ LD's.

* Regression :-

- ① It is a supervised learning technique.
- ② Deals with drawing a curve or line which minimizes sum of sq. errors



$$\text{③ error} = \|\hat{y} - y_i\|_2^2$$

$$\text{sum sq. error} = \sum \|\hat{y} - y_i\|_2^2$$

$$\text{Mean sq. error} = \frac{\text{sum sq. error}}{N}$$

$$\underset{m, c}{\text{minimize}}, E = \sum_{i=1}^n \|y_i - mx_i - c\|^2$$

$$\therefore \frac{\partial E}{\partial m} = \sum_{i=1}^n -2(y_i - mx_i - c)x_i = 0$$

&

$$\frac{\partial E}{\partial c} = -2 \sum_{i=1}^n (y_i - mx_i - c) = 0$$

$$\therefore \sum_{i=1}^n (y_i - mx_i - c)x_i = 0 \quad -\textcircled{1}$$

$$\sum_{i=1}^n (y_i - mx_i - c) = 0 \quad -\textcircled{2}$$

$$\therefore \sum y_i x_i - m \sum x_i^2 - c \sum x_i = 0$$

&

$$\sum y_i - m \sum x_i - cn = 0$$

$$\therefore m \sum x_i^2 + c \sum x_i = \sum y_i x_i$$

&

$$m \sum x_i - nc = \sum y_i$$

This can be written as

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

$$\therefore \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix}^{-1} \begin{bmatrix} \sum y_i x_i \\ \sum y_i \end{bmatrix}$$

This is called least square regression.

If $Ax=b$ is a system, then $b-Ax$ is error &

$$\|b-Ax\|_2^2 = \text{least square error}$$

\therefore we want to minimize $(b-Ax)^T(b-Ax)$

$$= (x^T A^T - b^T) (b - Ax)$$

$$= (x^T A^T \hat{Ax} - CAx)^T b - b^T A x + b^T b.$$

$$\Rightarrow A^T A \hat{x} - A^T b = 0$$

$$\therefore A^T A \hat{x} = A^T b$$

$$\therefore \hat{x} = (A^T A)^{-1} A^T b$$

This is called least square fitting

$(A^T A)^{-1} A^T$ \Rightarrow Pseudo inverse or
Moore's-penrose inverse.

Also we can write,

$$\begin{bmatrix} x_1 & | & \\ x_2 & | & \\ \vdots & | & \\ x_n & | & \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

which is in $Ax = b$ form. then we
can use A & find \hat{x} using
 $\hat{x} = (A^T A)^{-1} A^T b$ to find approximate
solution.

$$\min_{\mathbf{x}} \|\mathbf{b} - A\mathbf{x}\|^2$$

$$\text{sub. to } \|\mathbf{x}\|_2^2 = 1$$

- ① This is called L^2 -regularization problem or ridge regression problem.
- ② It is a constrained minimization problem.

Also, $L(\mathbf{x}, \alpha) = \min \left\{ \|\mathbf{b} - A\mathbf{x}\|_2^2 - \alpha \|\mathbf{x}\|_2^2 \right\}$
 this lagrangian eqⁿ can be used for minimization.

$$\therefore \frac{\partial L}{\partial \mathbf{x}} = \mathbf{x}^T A^T A \mathbf{x} - 2 \mathbf{x}^T A^T \mathbf{b} - 2\alpha \mathbf{x}^T \mathbf{x} + \mathbf{b}^T \mathbf{b}$$

$$0 = 2A^T A \mathbf{x} - 2A^T \mathbf{b} - 2\alpha \mathbf{x}$$

$$\therefore \hat{\mathbf{x}} = (A^T A - \alpha I)^{-1} A^T \mathbf{b}$$

This is called parameterized Reg. Model or Tikhonov regression. or least square regression model.

The above solution is generally used to remove noise from data.
i.e. using

$$\hat{x} = (A^T A - \alpha I)^{-1} A^T b$$

If $\alpha = 0 \Rightarrow$ Normal L_2 regularization problem.

more $\alpha \rightarrow$ less error i.e. $\min. ||\hat{x} - x_i||_2^2$
i.e. data will be denoised but we might loose some important texture & finer details of image.

So we should choose ' α ' in such a way that important data will not be lost & maximum noise get removed.

If data has texture \Rightarrow go for L_1 regularization
If data doesn't has texture \Rightarrow go for L^2 regularization
(or data is smooth)

edges \rightarrow continuous discontinuities.

choice of regularization is heuristic
i.e. trial & error.

$$\min x, \quad \|b - Ax\|_2^2 \quad \text{s.t.} \quad |x| = 1$$

This is a L' regularization problem.

$$\therefore L(x, \alpha) = \min \left\{ \|b - Ax\|_2^2 - \alpha|x| \right\}$$

$$\therefore \frac{\partial L}{\partial x} = 0 \Rightarrow A^T A x - \alpha \frac{x}{|x|} = A^T b$$

$$\therefore \left(A^T A - \frac{\alpha}{|x|} \right) x = A^T b$$

not invertible
directly
because of $|x|$

Here we cannot directly take inverse
of bracket term. We need to go for
iterative soln. & It is called **Iterative
regularization problem.**

