✯ Linear Transforms

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

& $T \rightarrow$ 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, Transaction, Reflection, etc.

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

$b \rightarrow$ output vector

& $A \rightarrow$ Transformation Matrix

Fundamental subspaces of vectors :-

1) Column space
2) Row space. (column space of transpose)
3) Null space
4) left Null space. (Null space of (matrix)^T)

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

* 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 $N(A) \neq \phi \Rightarrow |A| = 0$

18/2/2021

★ MATRICES

To check if $Ax = b$ has a solution

(i) check $S(A) = S(A|b)$

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

inverse Operation

- well posed system
- 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 → operator
x → original image
b → 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 ⇒ well conditioned system

if $C_N >> 1$ ⇒ ill conditioned system

↳ 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 ⇒ ill conditioned

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

Diagonal Dominance :-

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

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

Gauss elimination :-

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

prod. (dia(U)) = |A| ← IMP

* LU Algorithm :-

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

$U = A$

```
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 ⇒ orthonormal matrix
(iv) R ⇒ upper Dinangular matrix
(v) Q will always have left inverse.
(vi) If A is square, Q will also have right inverse.

(v) ⇒ $Q^T Q = I$

(vi) ⇒ $Q Q^T = I$

&

$$\boxed{Q^T A = R}$$

If A is square then Q is called unitary matrix.

i.e. $\boxed{Q^T Q = Q Q^T = I}$

Normed Linear spaces :-

Norm ⇒ length of a vector.

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

Properties of Norm :-

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

* Inner product:

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

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

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

* orthonormality ⇒

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

norm of orthonormal vector = 1

* For any orthogonal matrix, left inverse always exists.

i.e. $\boxed{Q^T Q = I}$

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

i.e. $\boxed{Q^T Q = Q Q^T = I}$

Properties of unitary transformation.

① unitary transformations preserves length of vector.

i.e. given a, Q, unitary matrix

& $Qx = y$ 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 ⇒ in unitary transformation, corelated data becomes uncorelated.

Gram-Schimidt Orthogonalization:

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

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

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.

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

Eigen Decomposition:-

Any sq. mat A can be decomposed into

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

$\Sigma \rightarrow$ 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, $\boxed{A^{-1} = A^T} \Rightarrow \boxed{X^{-1} = X^T}$

$$\therefore \boxed{A A^T = A^T A = I}$$

$\therefore$ A is unitary.

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

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

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

∴ For symmetric A, we have

$$X^{-1} = X^T \quad \& \quad A = X \Sigma X^T$$

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.

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

* positive Definite matrix :-

$\langle x, Ax \rangle$ i.e. dot product of x & Ax.

① if $X^T A X \geq 0$ & $\bar{x} \neq 0$, then A is called, positive semi-definite. (i.e. $\forall i$ $\lambda_i \geq 0$)

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

i.e. $\forall_i$, $\lambda_i > 0$

SPD matrices :-

(symmetric positive definite)

a spd matrix A can be decomposed into

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

SPSD → symmetric positive semi-definite

then if A is SPSD,

we have,

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

is done by removing all zero eigen values.

By doing this, we avoid ill-condition-edness & This is called precondition-ing the matrix. This avoids ill-conditioned system.

* preconditioning a matrix ⇒ 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 = B

i.e. $$AB = B^{-1}(BA)B$$

Also A^TA & AA^T are similar & ∴ 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
& S → diagonal matrices.

- Unitary matrix → performs rotation
  preserves length

- Diagonal matrix → performs scaling
  preserves orientation

∴ SVD ⇒ Rotation-scaling-rotation

Now, $A = U \Sigma V^T \longrightarrow$ SVD

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

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

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

$$\therefore \boxed{A A^T = U \Sigma^2 U^T}$$

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

$$\tilde{A} = \sum_{i=1}^{k} 6_i v_i v_i^T$$

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

Also, if U_K & V_K are truncated matrices then.

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

* 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.

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

↑ principle component matrix.

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

then M = column mean = 2, 3

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

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

$\therefore$ 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

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

eigen vectors of $C \Rightarrow$ principle component

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

where, $S = diag(6_1, 6_2, 6_3, \ldots)$

$\underset{n\times n}{U}$ = eigen vectors of C = Principle component

we want to maximize $x^T C x$

where

x - turns out to be eigen vectors of C.

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

* Principle component are ⊥er to eachother.

* principle component are ⊥er to one another.

* Preservation of data variation:-
Principle component analysis preserves variation in data.

$\hat{x} = U U^T x$ → also $\hat{x} = x V V^T$ → reconstruction of data.

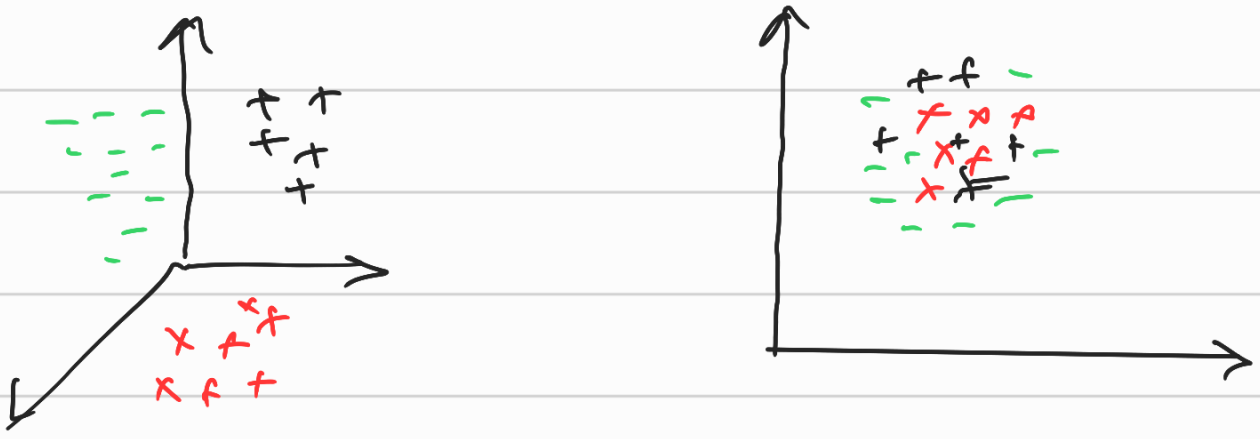
where $\hat{x}$ is projection of x from higher dimension to lower dim.

- So, if we have k-prineiple components then they represent basis for k-dimensional space.
- Also, Any 'p'-dimensional veetor 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 = 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 ⟶ lower dimension.

* PCA is mainly used for dimension reduction.

* Kernel - PCA

Used to convert data from lower dimension (x) to higher dimension space (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 $e^{-\|x-y\|_i^2/\sigma^2} \Rightarrow$ 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 $x_1, x_2, \ldots, x_n$ in dataset & we want to project x_0, then it is called out off sample projection & It can be found as follows

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

* out of sample reconstruction :-

as, $\tilde{x} = U U^T x_0$

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

But we cannot compute $\tilde{x}$ 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) :-

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

* $K \times \left[I - \frac{1}{N}(1\,1^T)\right] \Rightarrow$ This subtracts row mean from corr. rows. $\Rightarrow$ centralizing in terms of rows

where $1 \rightarrow$ column vector with no. of rows = no. of rows in K.

&

$N \Rightarrow$ No. of samples / No. of colums 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 f^n form,

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

$\lambda \rightarrow$ Lagrangian parameter.

Now, $\frac{dL}{dx} = 0 \Rightarrow \frac{d}{dx}\left[x^T S x - \lambda x^T x\right] = 0$

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.

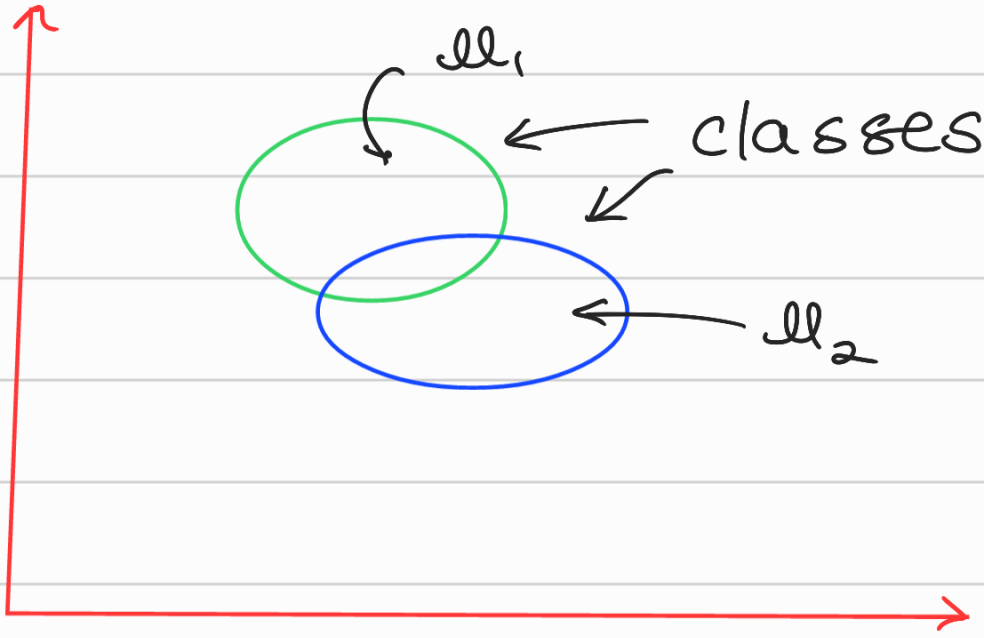
④ In LDA we find vector for classification of data & in PCA we find vector for dimension reduction

***

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-1 LD vectors)

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

$$\underset{\omega}{\text{maximize}}, \quad ||(\mu_1 - \mu_2)^T W||_2^2$$

i.e. $\max \left((\mu_1 - \mu_2)^T W\right)^T \left((\mu_1 - \mu_2)^T \omega\right)$

$\therefore \max \left(\omega^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \omega\right)$

where, $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

= between class scatter matrix.

& S_ω = within class sc. matrix.

i.e. $\max. \quad \omega^T S_B \omega$

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

where, $S_\omega = \Sigma_1 + \Sigma_2$

& $\Sigma_1 = (x_1 - \mu_1)(x_1 - \mu_1)^T \rightarrow$ for class 1

$\Sigma_2 = (x_2 - \mu_2)(x_2 - \mu_2)^T \rightarrow$ for class 2

In turn,

$$\max_{\omega}, \quad \boxed{\frac{\omega^T S_B \omega}{\omega^T S_\omega \omega}} \rightarrow \text{Reigleigh's method}$$

or $L(\omega, \lambda) = \max (\omega^T S_B \omega - \lambda \omega^T S_\omega \omega)$

↳ Lagrange's method.

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

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

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

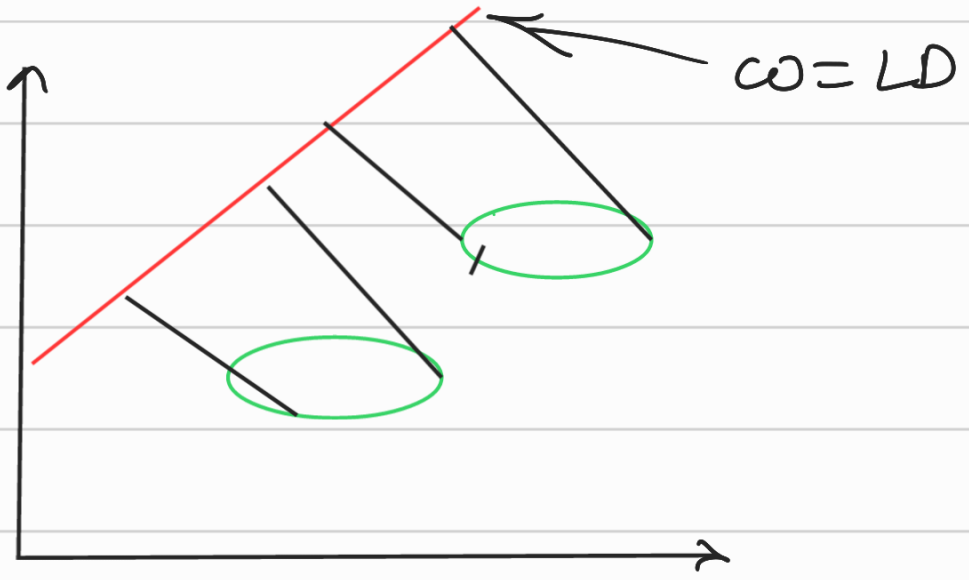
$$\therefore \quad (S_\omega^{-1} S_B)\, \omega = \lambda \omega$$

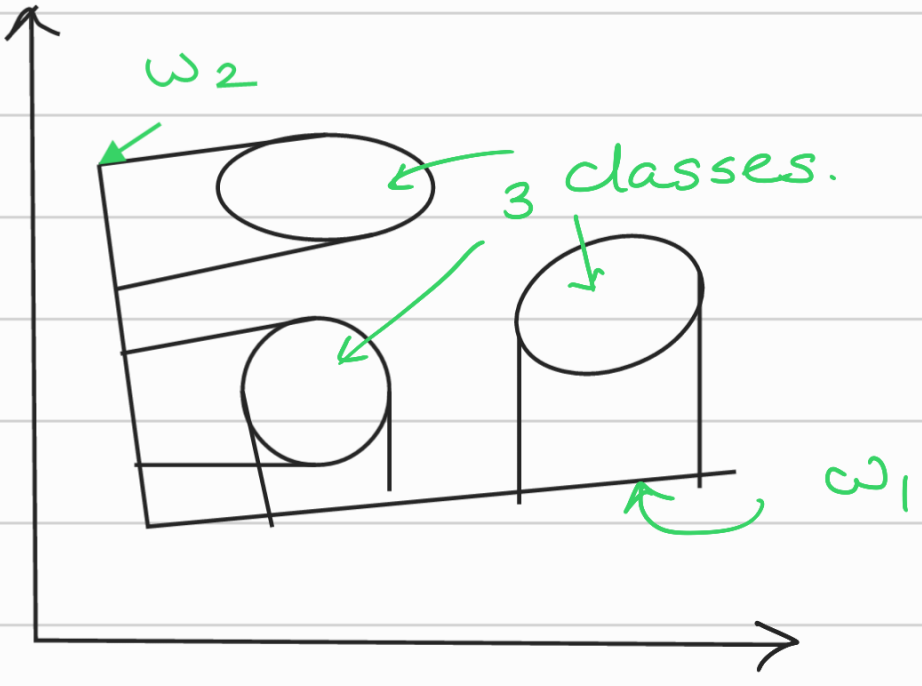
which is similar to $Ax = \lambda x$

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

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

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



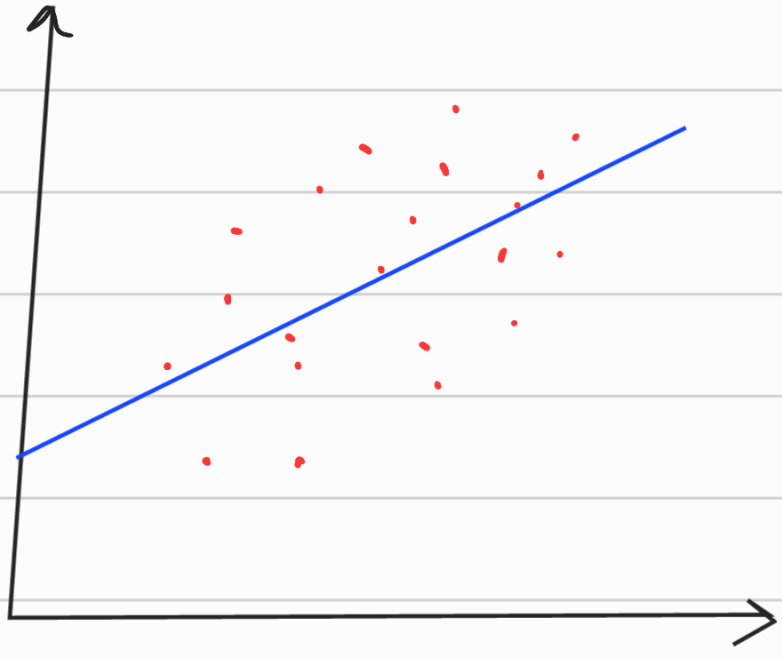


∴ for n classes we need n-1 LD's.

* Regression :-

① It is a supervised learning technique.

② Deals with drawing a curve or line which minimizes sum of sq. errors



③ $error = ||\hat{y} - y_i||_2^2$

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

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

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

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

&

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

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

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

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

&

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

$$\therefore \boxed{\begin{array}{c} m \sum x_i^2 + c \sum x_i = \sum y_i x_i \\ \& \\ m \sum x_i - nc = \sum y_i \end{array}}$$

This can be written as

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

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

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 $(b-Ax)^T(b-Ax)$

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

$$= (x^TA^TA\hat{x} - (Ax)^Tb - b^TAx + b^Tb.$$

$$\Rightarrow A^TA\hat{x} - A^Tb = 0$$

$$\therefore \boxed{A^TA\hat{x} = A^Tb}$$

$$\therefore \boxed{\hat{x} = (A^TA)^{-1}A^Tb}$$

This is called least square fitting

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

Also we can write,

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

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

$$\min_x \|b - Ax\|^2$$

sub. to $\|x\|_2^2 = 1$

① This is called L^2-regularization problem or ridge regression problem.

② It is a constrained minimization problem.

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

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

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

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

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

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

i.e. using

$$\hat{x} = (A^TA - \alpha I)^{-1}A^Tb$$

If $\alpha = 0 \Rightarrow$ Normal L_2 regularalization 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 & maximium noise get removed.

If data has texture $\Rightarrow$ go for L^1 regularization

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

edges → continuous discontinuities.

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

$$\min_x \; \|b - Ax\|_2^2 \quad s.t. \quad |x| = 1$$

This is a L^1 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.