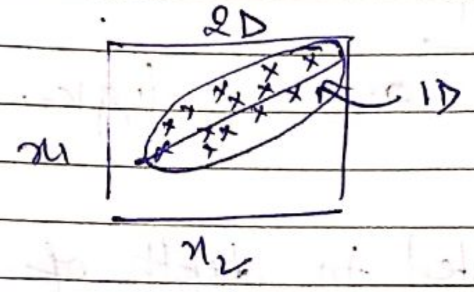
DIMENSION REDUCTION

Uses.

→ To Reduce Computational Cost.
→ To remove redundant dimension.
→ some dimension, info is not significant, variance is low / negligible
→ Data Visualization in lower dim.



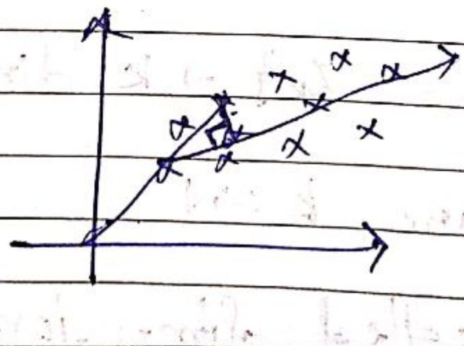
Principle Component Analysis. (PCA)

$\underbrace{N \longrightarrow K}$

projection / Principle Component.

Project data only along the direction of maximum variance.

→ Try to Capture maximum variation in N dimension while transforming to k-dimension.



Page No.
Date

Projection line is Vertical line from data pts to fitted straight line.

Angle to the fitted str line that gives shortest Projection error.

PCA Method / Algorithm

→ Mean Normalization $\mu = 0$, $\sigma = 1$
→ Feature Scaling: to bring dimensions to similar range.

→ Co Variance Matrix
$C_1 = X^T X = (N \times N)$ dim

Eigen Analysis.

Cov mat: Σ be an $N \times N$ matrix

λ be eigen value of Σ if $p \neq 0$ vector

$\Sigma p = \lambda p$. λ is scalar

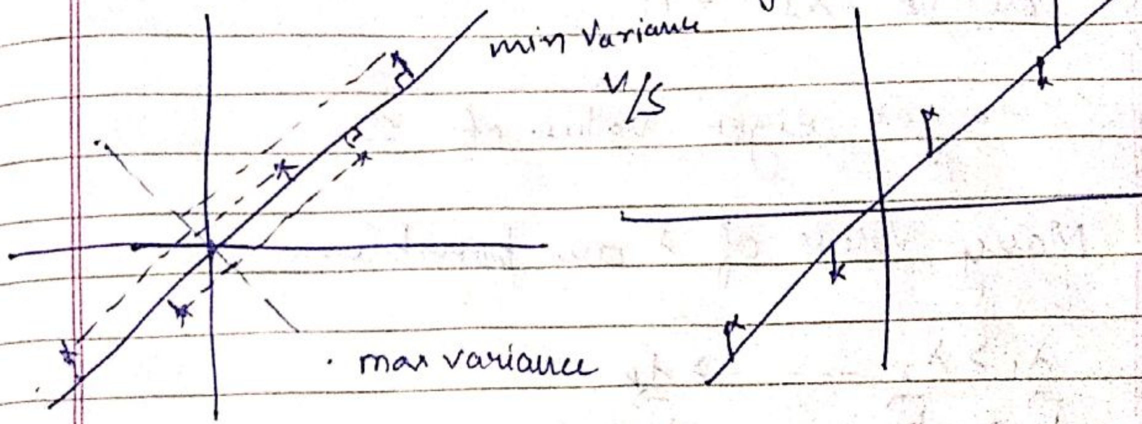
$$\begin{bmatrix} \\ \\ \end{bmatrix}_{N \times N} P_{N \times 1} = \lambda P_{N \times 1}$$

Solve for Eigen values
$\det(\Sigma - \lambda I) = 0$

Dimentionality Reduction
(Cont...)

PCA

Principle Component Analysis.



PCA — Linear Regression.

2 dim → 1 dim   diff

~~fitting data~~ :

Eigen Analysis

Let Σ (Covariance Matrix) be $N \times N$ matrix

$X = M \times N$ dimensional.

$$\underset{N\times M \;\; M\times N}{\Sigma = X^T X} = [N \times N].$$

↓

Variation b/w dimensions of X

$$\vec{\Sigma}\vec{p} = \lambda \vec{p}$$

λ → Eigen Value

$\vec{p}$ → Eigen Vector

$$\Sigma p - \lambda p = \vec{0}$$
$$(\Sigma - \lambda I)\, p = \vec{0} \qquad p \neq 0 \quad p \text{ is non-zero vector}$$

$$\det|\Sigma - \lambda I) = 0$$

to get eigen values of Σ

Many values of λ are possible.

$\lambda_1 > \lambda_2 \; --- > \lambda_k$
$\vec{b}_1, \vec{b}_2, -- \vec{b}_k$

higher $\lambda \uparrow$ → More Variation Captured

SVD (singular value Decomposition.)

$[U, S, V] = SVD(\Sigma)$

Also Applicable for Rectangular Matrices

U S V

N×N

$$\underset{U}{\begin{bmatrix} U_{11} & u_{21} & -- & U_{N1} \\ u_{12} & u_{22} & & | \\ | & | & & \\ u_{1N} & u_{2N} & & U_{NN} \end{bmatrix}} \underset{S}{\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ 0 & & \lambda_3 & \\ & & & \lambda_N \end{bmatrix}} \underset{V}{\begin{bmatrix} V_{11} & & V_{N1} \\ V_{12} & & \\ | & & \\ V_{1N} & & V_{NN} \end{bmatrix}}$$

projection vector
Eigen vector

λ_1,
Eigen value.

In SVD we consider only first k projection vector for Dimensionality Reduction.

How choose K?

K cols in U matrix. Resulting U_K matrix.

$X = M \times N$.

$Z = X U_K$
$= M \times N \quad N \times K$

$\tilde{X} = M \times K$.

Reduction from $N \rightarrow K$.

Sq. projection error $= \frac{1}{m} \sum_{i=1}^{M} \| x^i - \tilde{x}^i \|^2$

Total Variation $= \frac{1}{m} \sum_{i=1}^{M} \| x^i \|^2$

choose k such that

$$\frac{\frac{1}{m} \sum_{i=1}^{M} \| x^i - \tilde{x}^i \|^2}{\frac{1}{m} \sum_{i=1}^{M} \| x^i \|^2} \leq 0.01$$

If Variance Captured is $\simeq 99\%$ for smallest k we consider that k

99% variance is modelled

$$\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{N} S_{ii}} \geq 0.99.$$

Classification Using PCA

⇒ Done for Data Compression.

Face Recog.

M training eg → N dim.

$N \rightarrow k$ a $k << N$

$\Sigma \rightarrow N \times N$ → Variance accross diff dimension.

$$\underset{N \times N}{\Sigma} \underset{\times N\times 1}{p_k} = \lambda p_k^{N \times 1}.$$

$$x_k = \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \cdots & \tilde{x}_{1k} \\ \tilde{x}_{21} & \cdots & & \tilde{x}_{2k} \\ \tilde{x}_{M1} & \cdots & & \tilde{x}_{Mk} \end{bmatrix}$$

$$\tilde{x}_{11} = \vec{x}_1 \cdot \frac{\vec{p}_1}{\|p_1\|}$$

=

Page No.
Date

$\rightarrow$ Reduced Dim. Space.

$\vec{X}_k = \vec{X}\vec{p}_k$

$p_k = [\vec{p}_1 \ \vec{p}_2 \ -- \ \vec{p}_k]$.

Q.

$\tilde{X} \rightarrow$ classifier. (SVM/LR/Random forest)

Σ

$X_i \rightarrow 1\times N \rightarrow PCA \rightarrow \tilde{X}_i \rightarrow 1\times M$

$X_j \rightarrow 1\times N \rightarrow PCA \rightarrow \tilde{X}_j \rightarrow$ Classifier.

$\rightarrow$ In $M\times N$

X $M\times N$ dim

Σ $N\times N = X^T X$.

$\Sigma p_k = \lambda_k p_k$ $k = 1, 2 \ldots K$. p is $N\times 1$ dim

$X\Sigma p_k = \lambda_k X p_k$.

$X X^T X p_k = \lambda_k X p_k$

$L = X X^T$

$L X p_k = \lambda_k X p_k$. $\Rightarrow$ Eigen Equation

L:- Variance b/w data pts..

$q_k / X p_k$ is eigen Vector for 'L'

$q_k = X p_k$

L v/s Σ

$X X^T$ $X^T X$

(Not transpose)

Variance b/w data pts. (training eg.)

Variance b/w dimension

$k \leq M$, Let $k = M$

q_k, model variations in $M \times M$ Subspace which is much smaller than that of $N \times N$

Given a image we project $1 \times N$ to $M \times M$ ~~dimension~~. Subspace using q_k eigen vector to get $1 \times M$ dimension.

$W = q_k^T \times X_i^T$ $(M \times 1) = (M \times M) \times (M \times 1)$

W indicates contribution of each eigen Vectors. i.e. weight with each eigen vector

→ For each training Images find out w.

→ There are multiple images per class, Accordingly Multiple W

→ These act like feature ~~fe~~-vector. You may further process them to build classifier for each class. Otherwise store them as it is.

→ During Testing Y is processed by P_k to get $1 \times M$ vector from $1 \times N$

→ The compressed vector is further processed by q_k to get w of dimension $1 \times M$

→ This is compared with training w using knn or euclidian distance to find class

$$X_{M\times N} \xrightarrow[P_k]{PCA} \underset{M\times k}{X_k} \xrightarrow{q_k} W_{M\times k}$$

$$\downarrow$$

$$W_{M\times M}$$

$X_1 \rightarrow W_1$

$X_2 \rightarrow W_2$ ← KNN.

X_3 ⋮ W_3

X_4 ⋮ W_4

⋮

X_M W_k.

LDA

Linear Discriminant Analysis.

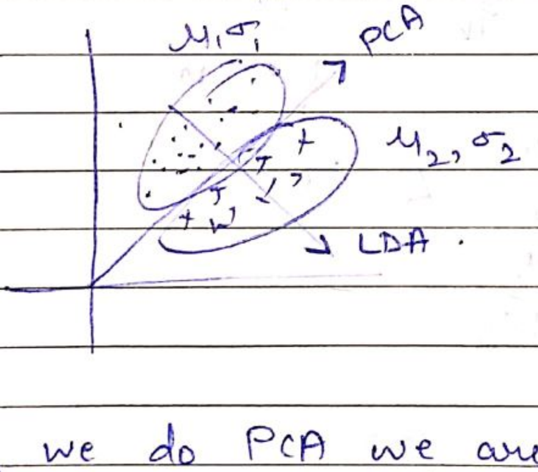
→ Also termed as fisher discriminant Analysis

→ PCA does Dimentionality Reduction by focussing on dimension having most variation

→ It may not of be of use when features among different classes lie along same direction of variance

→ Can we have approach to increse Separability between classes

→ PCA is unsupervised
→ LDA is Supervised.



PCA v/s LDA

If we do PCA we are mixing data of Different Class., then it won't of much use.

In LDA
$(\mu_1 - \mu_2)$ maximization.

σ's we will reduce separately.
Intraclass variance reduction
Interclass mean maximization.

How LDA creates new axis?

→ New Axis is created according to 2 criteria.
→ maximize $\mu_1 - \mu_2$
→ minimize intra-class variance

ratio $\dfrac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} = \dfrac{d^2}{\sigma_1^2 + \sigma_2^2}$

for better separation among classes, numerator must be as large as possible, while denominator must be as small as possible.

LDA in case of 2 Class

Consider $X \Rightarrow M \times N$

Goal:- project this data onto one dimension & at same time maximize separability among 2 classes.

x is one eg. $N \times 1$ & , W of dim $N \times 1$

$Z = W^T x$.

Let μ_1, μ_2 are mean of 2 class.

Projected means are $W^T \mu_1$ & $W^T \mu_2$

Σ_1, Σ_2 are Co-variance of 1, 2

Projected cov $W^T \Sigma_1 W$ & $W^T \Sigma_2 W$

Let F = Euclidian distance b/w projected mean

$$F = \| W^T \mu_1 - W^T \mu_2 \| = (W^T \mu_1 - W^T \mu_2)^T (W^T \mu_1 - W^T \mu_2)$$
$$= W^T (\mu_1 - \mu_2)(\mu_1^T - \mu_2^T) W$$

$F = W^T \Sigma_B W$

$\Sigma_B = (\mu_1 - \mu_2)(\mu_1^T - \mu_2^T)$ is b/w class Covariance

first Goal maximize $W^T \Sigma_B W$

Second minimize $W^T \Sigma_1 W + W^T \Sigma_2 W$

$W^T \Sigma_W W$

$\Sigma_W = \Sigma_1 + \Sigma_2$

$$\max \frac{W^T \Sigma_B W}{W^T \Sigma_W W}$$

Minimizing using Langrange's Multiplier

$$L(w, \lambda) = (w^T \Sigma_B w) - \lambda (w^T \Sigma_W w - 1)$$

$$\frac{\partial L}{\partial w} = 2 \Sigma_B w - 2\lambda \Sigma_W w = 0$$

$$\Sigma_B w = \lambda \Sigma_W w$$

$$\Sigma_W^{-1} \Sigma_B w = \underbrace{\lambda}_{\text{Eigen value}} \underbrace{w}_{\text{Eigen Vector}}$$