ACDS Lecture Series

Lecture - 9

CSIR

Dimensionality Reduction

G. N. Sastry & Team

ADVANCED COMPUTATION AND DATA SCIENCES (ACDS) DIVISION

CSIR-North East Institute of Science and Technology, Jorhat, Assam, India

9. Dimensionality Reduction

Introduction

Dimensionality reduction is a process of deriving a set of degrees of freedom which can be used to reproduce most of the variability of a data set.

Manifold learning techniques can be used in different ways:

- ***Data dimensionality reduction:*** Produce a compact low-dimensional encoding of a given high-dimensional data set.
- ***Data visualization:*** Provide an interpretation of a given data set in terms of intrinsic degree of freedom, usually as a by-product of data dimensionality reduction.
- ***Preprocessing for supervised learning:*** Simplify, reduce, and clean the data for subsequent supervised training.

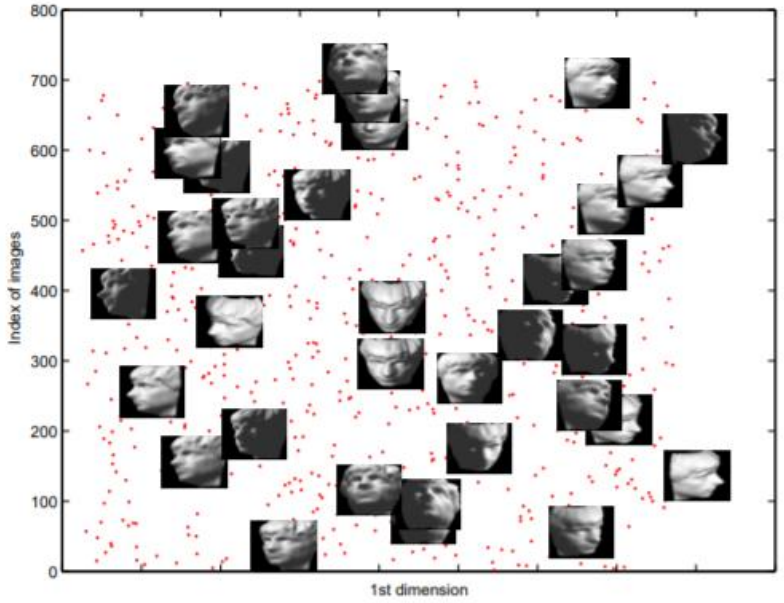


Figure 1. A canonical dimensionality reduction problem from visual perception. The input consists of a sequence of 4096-dimensional vectors, representing the brightness values of 64 pixel by 64 pixel images of a face. Applied to N = 698 raw images. The first coordinate axis of the embedding correlates highly with one of the degrees of freedom underlying the original data: left-right pose.

Features Space and Dimensions

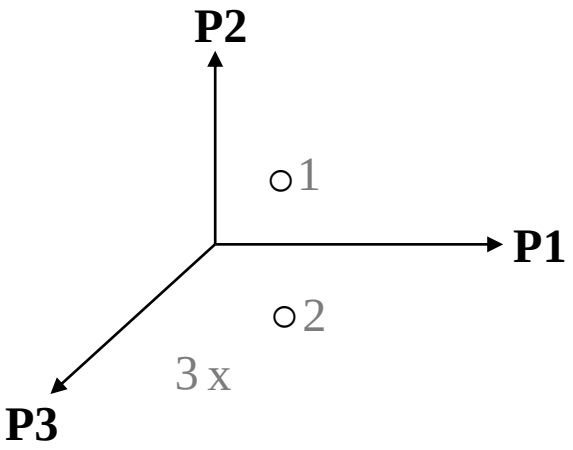
Let us consider the following example – Toxicity prediction (toxic or not?)

Features / Descriptors / Attributes | **Class**

Data Items / Instances

	Property1	Property2	Property3	Toxic or not?
Compound 1 →	a	b	9	yes
Compound 2 →	p	q	20	yes
Compound 3 →	x	y	-50	no

$$\begin{bmatrix} x_1^1 & x_2^1 & x_3^1 \\ x_1^2 & x_2^2 & x_3^2 \\ x_1^3 & x_2^3 & x_3^3 \end{bmatrix}_{3x3}$$



Features Space and Dimensions

Let us consider the following example – Toxicity prediction (toxic or not?)

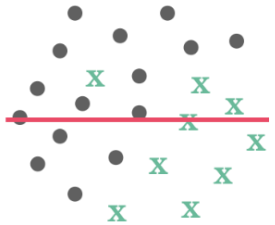
Features / Descriptors / Attributes | **Class**

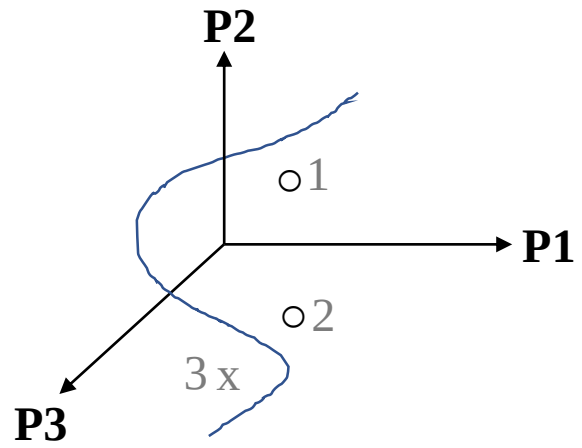
Data Items / Instances

	Property1	Property2	Property3	Toxic or not?
Compound 1 →	a	b	9	yes
Compound 2 →	p	q	20	yes
Compound 3 →	x	y	-50	no

Performance ?

Underfitting





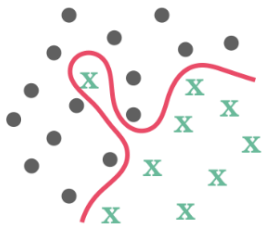
Features Space and Dimensions

Let us include more data items and more features to increase accuracy of the model -

	P1	P2	P3	P4	…	…	…	P97	P98	P99	P100	Toxic or not?
Compound 1 →												yes
Compound 2 →												yes
Compound 3 →												no
…												…
…												…
…												…
Compound 9998 →												no
Compound 9999 →												no
Compound 10000 →												no

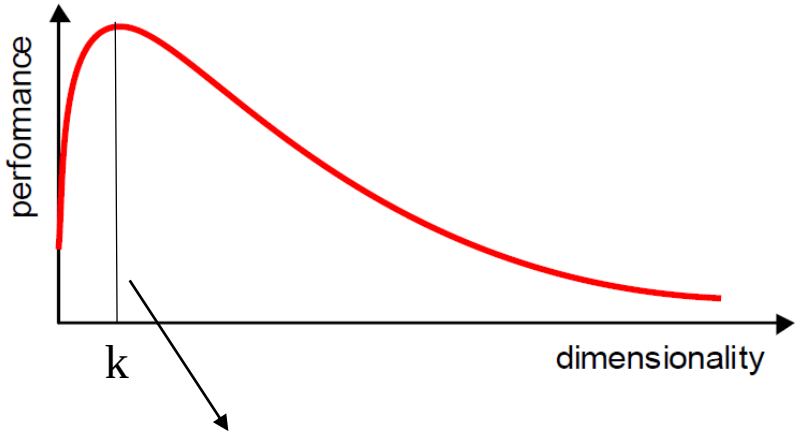
Overfitting

Performance ?

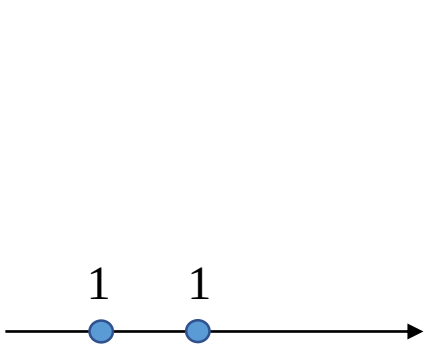


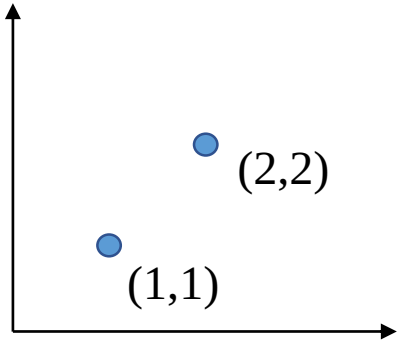
Curse of Dimensionality

- Increasing the number of features will not always improve classification accuracy.
- In practice, the inclusion of more features might actually lead to worse performance.



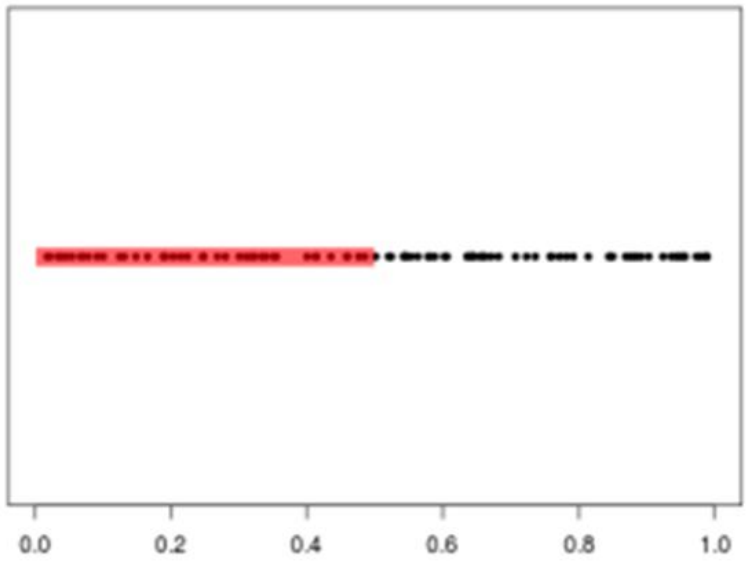
This is what (optimal) value we are looking for !!!

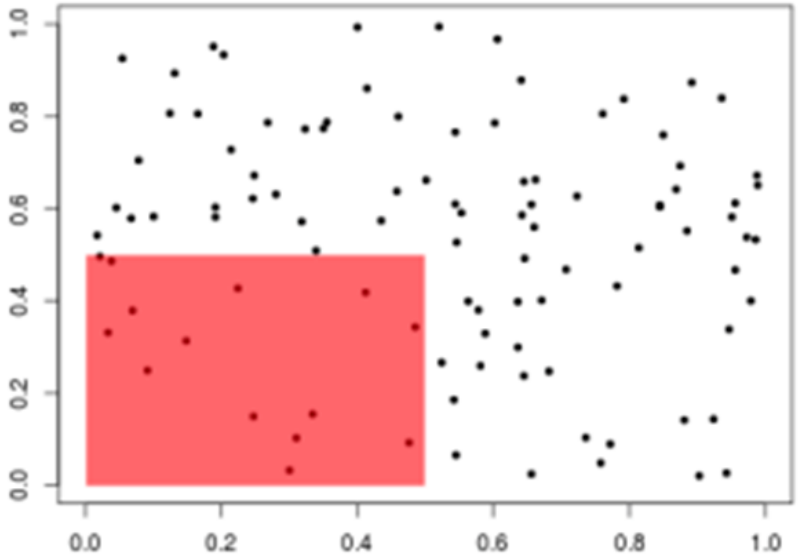


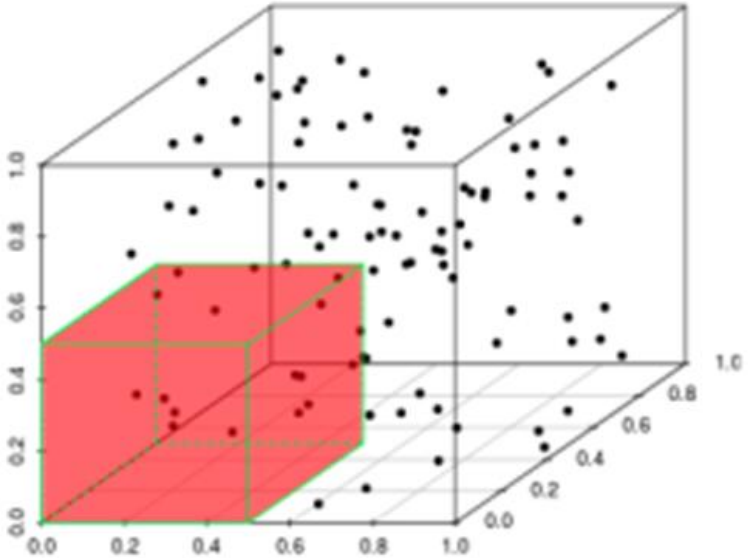


- As a result, high-dimensional datasets are at risk of being very sparse: most training instances are likely to be far away from each other.
- This also means that a new instance will likely be far away from any training instance, making predictions much less reliable than in lower dimensions, since they will be based on much larger extrapolations.

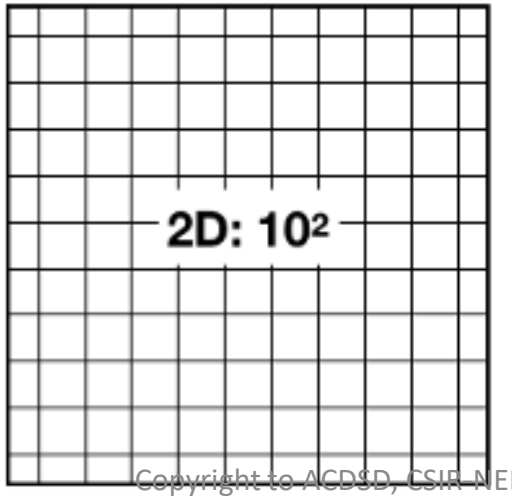
Curse of Dimensionality

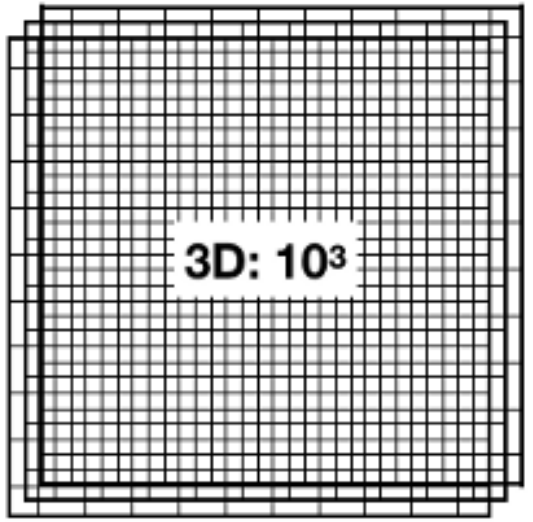






1D: 10^1





Dimensionality Reduction and Its Need

- Dimension reduction is the transformation/reduction of data from a high dimensional space into a lower dimensional space so that lower dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension.
- Very common in the fields that deal with large number of observations such as - Signal processing, Speech recognition, Bioinformatics and so on….

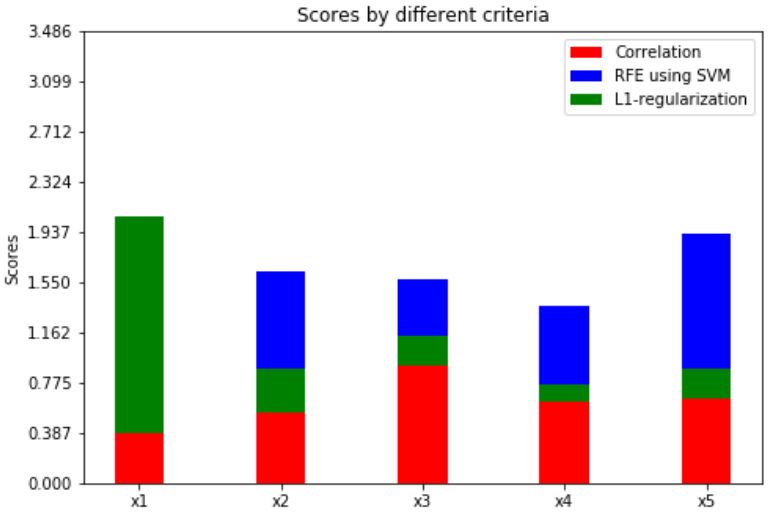
Advantages

- Alleviate curse of dimensionality
- Helps in visualizing data.
- Takes care of multicollinearity by removing redundant features.
- Reduced space complexity.
- Data analysis becomes computationally tractable.
- Some ML algorithms do not perform well when we have a large dimensions.

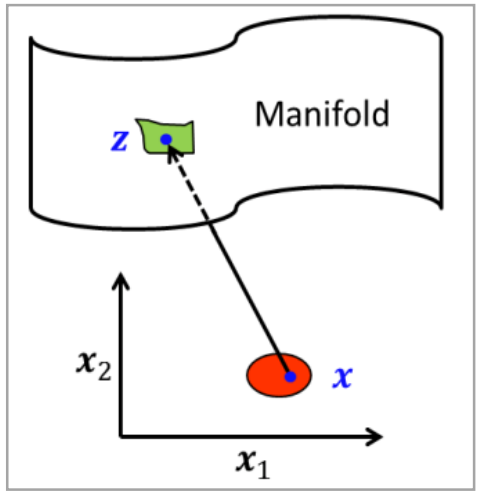
Techniques to Reduce Dimensions

Dimensionality Reduction

- Feature Selection/Elimination
- Feature Extraction

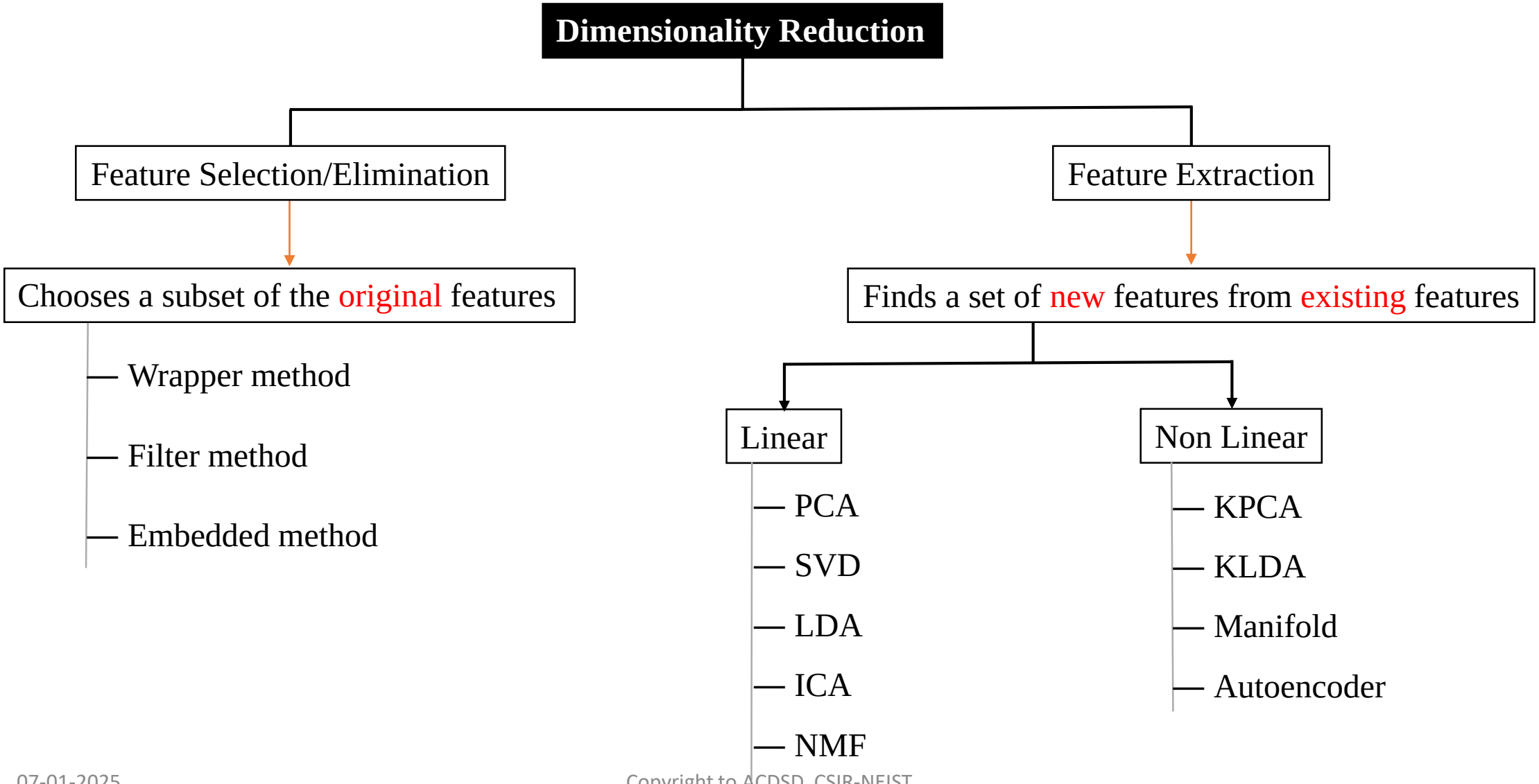


$$\begin{bmatrix} x_{11} & x_{21} & x_{31} & \dots & x_{n1} \\ x_{12} & x_{22} & x_{32} & & x_{n2} \\ x_{13} & x_{23} & x_{33} & \dots & x_{n3} \\ x_{14} & x_{24} & x_{34} & & x_{n4} \\ x_{15} & x_{25} & x_{35} & & x_{n5} \end{bmatrix}_{\mathbf{5\ x\ n}}$$



$$\begin{bmatrix} x_{1a} & x_{2a} & x_{3a} & \dots & x_{na} \\ x_{1b} & x_{2b} & x_{3b} & & x_{nb} \\ x_{1c} & x_{2c} & x_{3c} & \dots & x_{nc} \\ x_{1d} & x_{2d} & x_{3d} & & x_{nd} \\ x_{1e} & x_{2e} & x_{3e} & & x_{ne} \end{bmatrix}_{\mathbf{5\ x\ n}}$$

Techniques to Reduce Dimensions



Mathematics behind Dimensionality Reduction

Eigenvalue and eigenvector:

Eigenvalues and eigenvectors allow to "reduce" a linear operation to separate, simpler, problems.

A: an $n \times n$ matrix

λ: a scalar

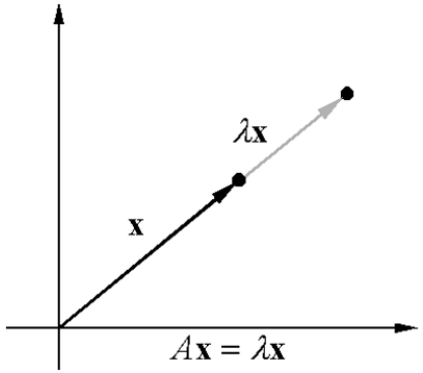
$\boldsymbol{x}$: a nonzero vector in R^n

Eigenvalue

$$Ax = \lambda x$$

Eigenvector

- Geometrical Interpretation



Transpose of Matrix

A Matrix which is formed by turning all the rows of a given matrix into columns and vice-versa.

If $A = [a_{ij}]_{m \times n}$ then $A^T = [a_{ij}]_{n \times m}$ i. e. consider the matrix below:

$$M = \begin{bmatrix} 2 & -9 & 3 \\ 13 & 11 & -17 \\ 3 & 6 & 15 \\ 4 & 13 & 1 \end{bmatrix}$$ hence the transpose of the matrix M is $$M^T = \begin{bmatrix} 2 & 13 & 3 & 4 \\ -9 & 11 & 6 & 13 \\ 3 & -17 & 15 & 1 \end{bmatrix}$$

- Inverse of Matrix

Any m x m square matrix M, which has zero determinant always has an inverse M^{-1}. It is mostly true for all the square matrix and is given by $MM^{-1} = M^{-1}M = I_{m.}$

Consider a matrix M = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ to compute the inverse of the matrix M we have to follow:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

determinant

Principal Component Analysis

- It is a classical method that provides a sequence of best linear approximations to a given high-dimensional observation.
- Given a set of data on “n” dimensions, PCA aims to find a linear subspace of dimension d lower than n such that the data points lie mainly on this linear subspace.
- The linear subspace can be specified by d orthogonal vectors that form a new coordinate system, called the ‘*principal components*’.
- **Hotelling explanation**: for a given set of data vectors xi , i ∈ 1...t, the d principal axes are those orthonormal axes onto which the variance retained under projection is maximal.

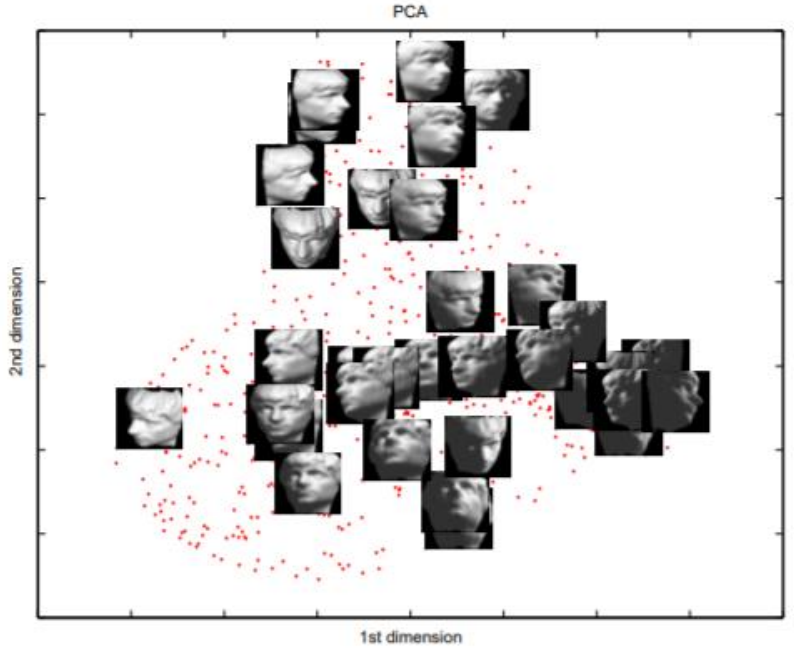
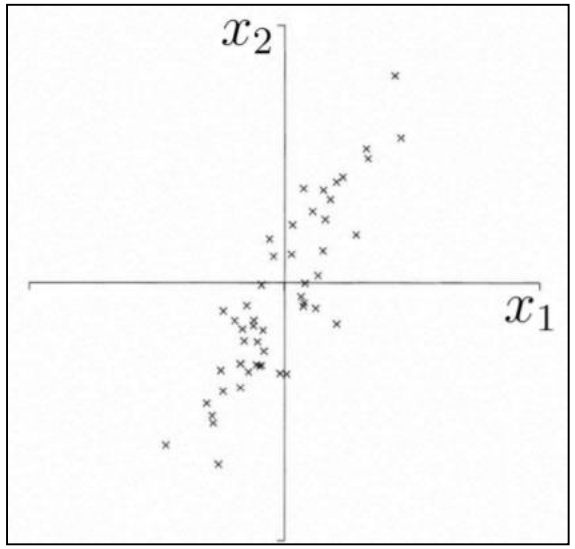


Figure 2. : PCA applied to the same data set. A two-dimensional projection is shown, with a sample of the original input images.

Principal Component Analysis

Principal component analysis (PCA) is a technique that is useful for the compression and classification of data. The purpose is to reduce the dimensionality of a data set (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample's information.

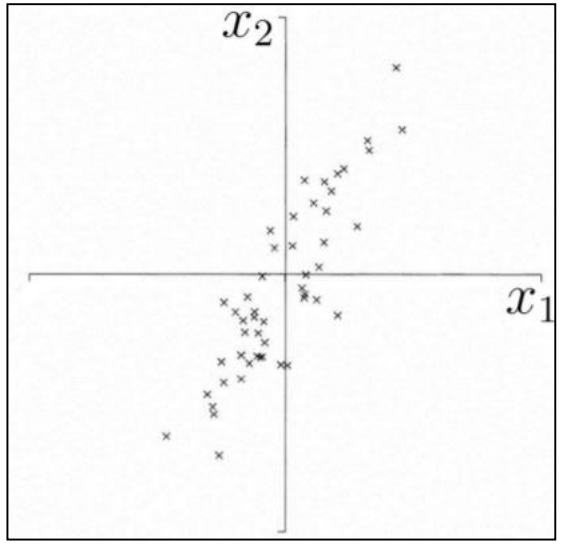
Geometric picture of principal components (PCs)

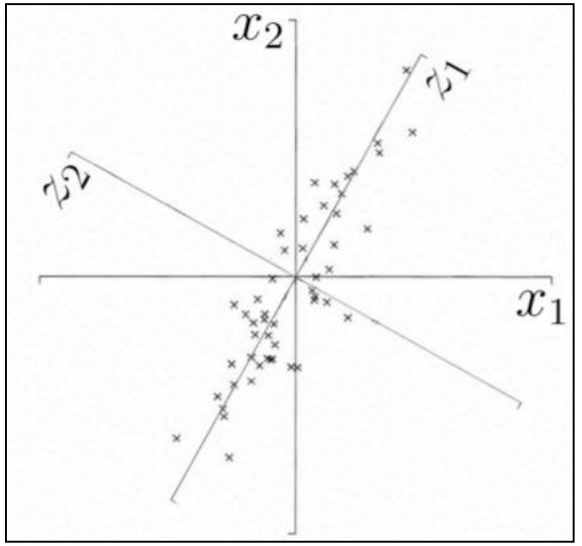


A sample of n observations in the 2-D space
X = (x1, x2)

PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous.

Geometric picture of principal components (PCs)

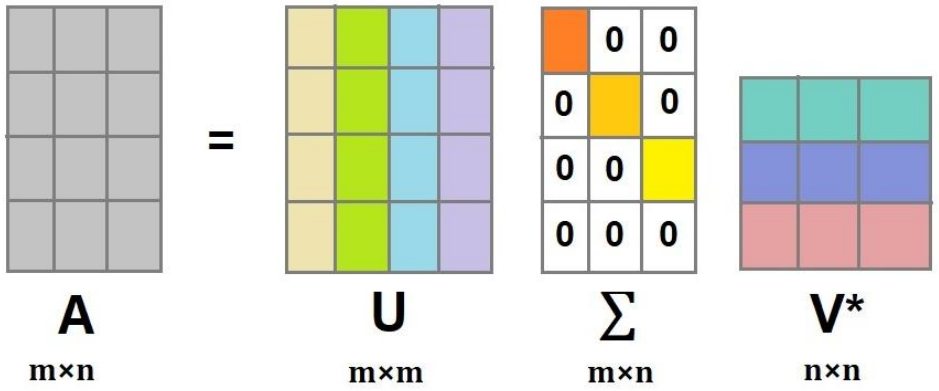


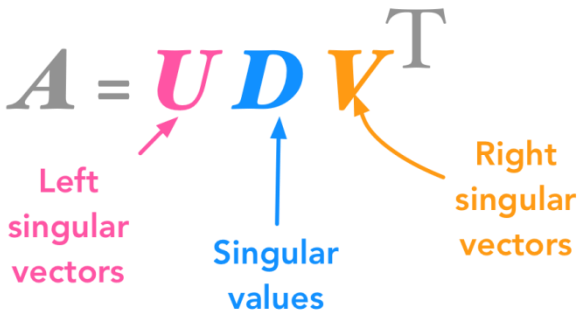


The 1st PC Z1 is a minimum distance fit to a line in X space
The 2nd PC Z2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC

Principal Component Analysis

Singular Value Decomposition





$$X_{c\,d\times n} = U_{d\times d}\,\Sigma_{d\times n}\,V^{T}_{n\times n}$$

i^{th} value Σ is the result of i^{th} column of U and i^{th} row of V

That means if we delete i^{th} column of U and i^{th} row of V
Then we can delete i^{th} value Σ

$$X_{c\,d\times n} = U'_{d\times k}\,\Sigma'_{k\times k}\,V'^{T}_{k\times n}$$

$$U'^{T}_{k\times d}\,X_{c\,d\times n} = \Sigma'_{k\times k}\,V'^{T}_{k\times n} = Y_{k\times n}$$

PCA accomplished

Generals steps to perform PCA:

i. Standardization of the dataset
ii. Computation of the covariance matrix
iii. Computing the eigen values & eigen vectors
iv. Computing the principal components
v. Reducing the dimensions of the data
vi. Representation of the PC on coordinate system

i. Standardization of the dataset:

Scaling our dataset in such a way that all the variables and their values lie within a similar range

Not doing so will result in biased output, inaccurate output

$$Z = \frac{Variable\ Value - Mean}{Standard\ Deviation}$$

ii. Computing the Covariance Matrix:

A covariance matrix denotes a correlation between different variables in a dataset. It is essential to identify heavily correlated/dependent variables because they contain biased & redundant information which reduces the overall performance of the model

A covariance matrix is represented as:

$$\begin{bmatrix} cov(a,a) & cov(a,b) \\ cov(b,a) & cov(b,b) \end{bmatrix}$$

Here, cov(a, a) means: covariance of 'a' with respect to 'a' ⇒ variance of 'a'
Here, cov(a, b) means: covariance of 'a' with respect to 'b'
Here, cov(b, a) means: covariance of 'b' with respect to 'a'
Here, cov(b, b) means: covariance of 'b' with respect to 'b' ⇒ variance of 'b'

❖Also, cov(a, b) = cov(b, a)

iii. Calculating the eigen values & eigen vectors:

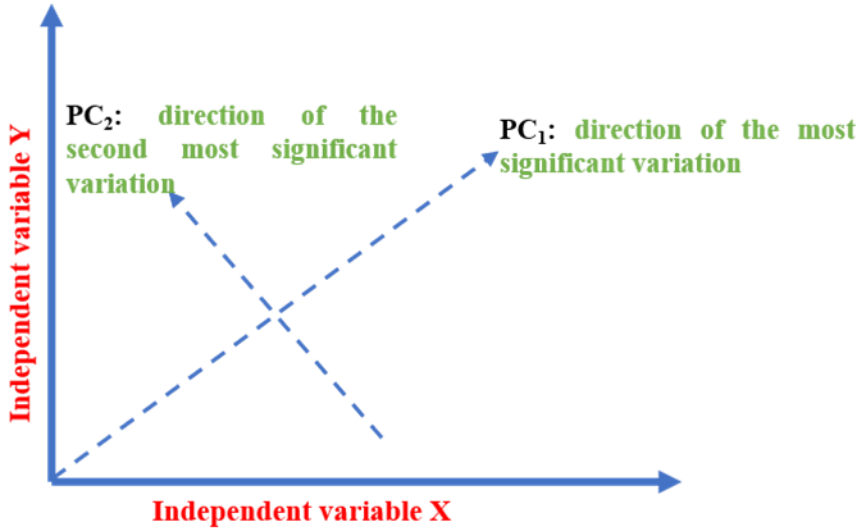
- Eigen values & eigen vectors are calculated using the covariance matrix and are used in the determination of principal components.
- A covariance matrix denotes a correlation between different variables in a dataset.
- The idea behind eigen vectors is to use the covariance matrix to understand as where in the dataset there is most amount of variance; because the covariance matrix gives us the overall variance and hence eigen values & eigen vectors are used to understand the variance in the dataset accordingly.
- Eigen vectors are used to identify as to where in the dataset we have the maximum variance; i.e., along which direction or corresponding to which variable or in what way we obtain the maximum variance in our dataset; as variance denotes containment of more information in that dataset and that is the main idea behind computing principal component.
- We need to calculate the PC because we need to store maximum information and maximum information is stored corresponding to maximum variance; that's the idea behind computing eigen values & corresponding eigen vectors.
- Eigen values are nothing but the scalar representation of eigen vectors.
- Eigen values & eigen vectors will compute the PC of the dataset.

iv. Computing the Principal Components:

- Once we have calculated the eigen values & corresponding eigen vectors, all we need to do is to order them in descending order
- While doing so, the eigen vector with highest eigen value is most significant and hence its PC_1, similarly, PC_2, PC_3, PC_4, …….and so on, from most significant PC towards least significant PC.

v. Reducing the Dimensions of the dataset:

This is the last step of PCA; in this step we rearrange the original data with respect to the final PC that we have obtained, which represents the maximum & most significant information of the dataset.



Principal Component Analysis

Choosing Number of Principal Components (k that explains most variance)

Avg. squared projection error: $\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)} - x_{approx}^{(i)}\|^2$

Total variation in data: $\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}\|^2$

Choose k s.t. (if 99% variance to be retained): $\frac{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)}\|^2} \leq 0.01$

PCA Implementation Variants

Randomized PCA

- Uses a stochastic algorithm called *Randomized PCA* that quickly finds an approximation of the first d principal components.
- Its computational complexity is $O(m \times d2) + O(d3)$, instead of $O(m \times n2) + O(n3)$ for the full SVD approach, so it is dramatically faster than full SVD when d is much smaller than n:

Incremental PCA

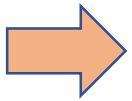
- One problem with the standard implementations of PCA is that they require the whole training set to fit in memory in order for the algorithm to run.
- *Incremental PCA* (IPCA) algorithm allow to split the training set into mini-batches and feed an IPCA algorithm one mini-batch at a time.

Advantages of PCA

- Improve Model Performance:

✓ Alleviate curse of dimensionality
✓ Helps in visualization
✓ Takes care of multicollinearity
✓ Reduce space and time complexity

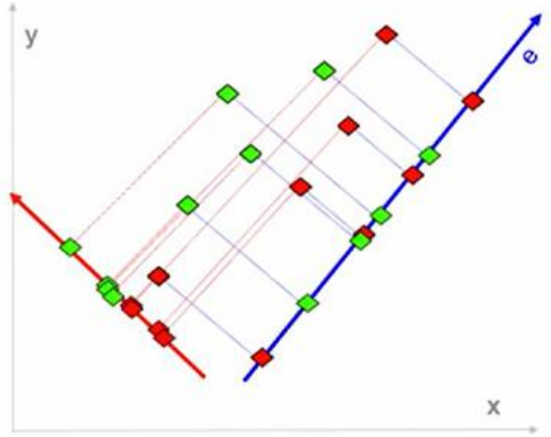
What About !!!
"Loss of Information"



Transformation into small dimension k can result in Information Loss

Principal Component Analysis

Shortcomings of PCA (standard)



- PCA is unsupervised
  - Maximizes overall variance of the data along a small set of directions (k)
  - Does not care anything about class labels
  - Thus can pick direction that makes it hard to separate classes

➢ Solution is go for a discriminative approach (FLDA)

- Looks for a dimension that makes it easy to separate classes

Example

Qu- For the following data use PCA to reduce the dim. from 2 to 1 :

feature	ex_1	ex_2	ex_3	ex_4
x	4	8	13	7
y	11	4	5	14

Data Set-

Step ① no. of features = $n = 2$

no. of samples = $N = 4$

Step ② Mean of variables-

$$\bar{x} = \frac{4+8+13+7}{4} = 8$$

$$\bar{y} = \frac{11+4+5+14}{4} = 8.5$$

Step ③ Computation of Covariance Matrix –

The ordered pairs are – (x,x) (x,y) (y,x) (y,y)

- Covariance of the ordered pair (x_i, x_j) :–

$$\text{Cov}(x_i, x_j) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

@ $x_i = x_j \Rightarrow$

$$\text{Cov}(x,x) = \frac{1}{N-1} \sum_{k=1}^{N} (x - \bar{x})^2$$

Now here, $\text{Cov}(x,x) = \frac{1}{4-1}\{(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\}$

$= 14$

$\text{Cov}(x,y) = \frac{1}{4-1}\{(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)\}$

$= -11$

$\because \text{cov}(x,y) = \text{cov}(y,x) = -11$

Now, $\text{cov}(y,y) = \frac{1}{4-1}\{(11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\}$

$= 23$

$\therefore$ the covariance matrix is $(S) = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$.

Step ① Eigen value, eigen vector & Normalized Eigen Vector

- Eigen Value – $|S-\lambda I| = 0$

$\Rightarrow \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} = 0$

$\Rightarrow (14-\lambda)(23-\lambda) - (-11)^2 = 0$

$\Rightarrow \lambda^2 - 37\lambda + 201 = 0$

for, $ax^2+bx+c=0$

$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$

$\lambda = 30.3849, 6.6151$

Now, $\lambda_1 > \lambda_2$

$\Rightarrow \lambda_1 = 30.3849$ (PC_1)

$\lambda_2 = 6.6151$ (PC_2)

Now, Eigen vector of λ_1 :–

$(S-\lambda_1 I)U_1 = 0$

$\begin{bmatrix} 14-\lambda_1 & -11 \\ -11 & 23-\lambda_1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$

$\Rightarrow \begin{bmatrix} (14-\lambda_1)u_1 - 11u_2 \\ -11u_2 + (23-\lambda_1)u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$\Rightarrow$

$$\begin{cases} (14-\lambda_1)u_1 - 11u_2 = 0 \\ -11u_1 + (23-\lambda_1)u_2 = 0 \end{cases}$$

} from these two eqns we need to find u_1 & u_2.

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda_1} = t$$

@ $t=1 \Rightarrow$ $u_1 = 11$, $u_2 = 14-\lambda_1$ $\Rightarrow U_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}$

$$\Rightarrow U_1 = \begin{bmatrix} 11 \\ 14 - 30.3849 \end{bmatrix}$$

$$\Rightarrow U_1 = \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$$

Now, we have to normalize U_1 :-

$$e_1 = \begin{bmatrix} 11/\sqrt{(11)^2 + (-16.38)^2} \\ -16.3849/\sqrt{(11)^2 + (-16.38)^2} \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

Unit eigen vector (or) Normalized Eigen vector

Similarly,

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

Step (5) Derive new dataset ie, reduce the dimension of the old dataset -

	Ex_1	Ex_2	Ex_3	Ex_4
PC_1	P_{11}	P_{12}	P_{13}	P_{14}

This is the new, reduced dataset.

Now we need to find out P_{11}, P_{12}, P_{13} & P_{14} individually and put them here and we'll get reduced dataset.

Example

$$P_{xiy} = e_i^T \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \end{bmatrix}$$

$$\Rightarrow \quad P_{11} = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix} = [0.5574 \quad -0.8303] \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$$

$$\Rightarrow \quad P_{11} = -4.3052$$

Similarly,

$$P_{12} = [0.5574 \quad -0.8303] \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix}$$

$$\Rightarrow \quad P_{12} = 3.7361$$

Similarly, $P_{13} = 5.6928$

$P_{14} = -5.1238$

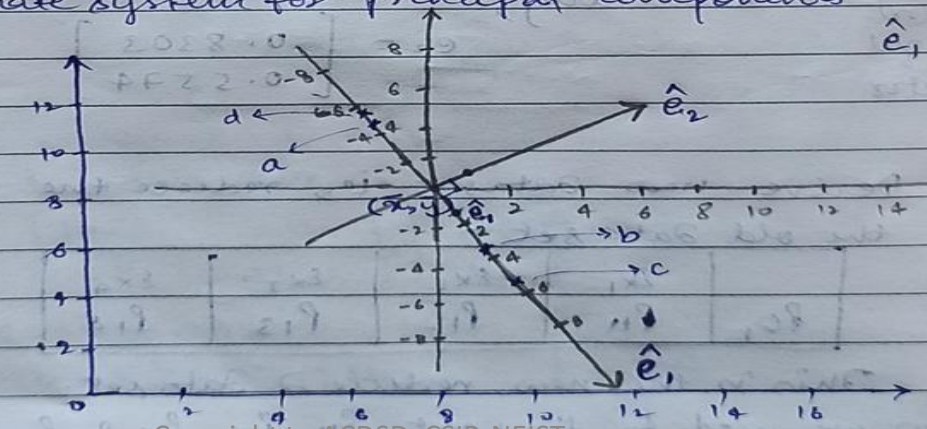
So the reduced data set is —

	Ex_1	Ex_2	Ex_3	Ex_4
PC_1	(a) -4.3052	(b) 3.7361	(c) 5.6928	(d) -5.1238

Here the reduced dim. is 1.

Now, Coordinate system for principal components —

$\hat{e}_1 \perp \hat{e}_2$



PCA is designed to model linear variabilities in high-dimensional data. However, many high dimensional data sets have a nonlinear nature. In these cases the high-dimensional data lie on or near a nonlinear manifold (not a linear subspace) and therefore PCA can not model the variability of the data correctly. One of the algorithms designed to address the problem of nonlinear dimensionality reduction is Kernel PCA.

Kernel PCA finds principal components which are nonlinearly related to the input space by performing PCA in the space produced by the nonlinear mapping, where the low-dimensional latent structure is, hopefully, easier to discover.

Consider a feature space $\mathcal{H}$ such that: $\Phi : x \to \mathcal{H}$

$$x \mapsto \Phi(x)$$

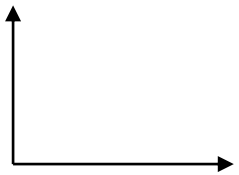
Kernel Variants

$$X = [x_1\ x_2\ x_3\ \ldots\ x_n]_{dxn}$$

$$X^T X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ . \\ x_n \end{bmatrix}_{nxd} [x_1\ x_2\ x_3\ \ldots\ x_n]_{dxn} = \begin{bmatrix} x_1^1 & \ldots & x_3^1 \\ x_1^2 & \ldots & x_3^2 \\ x_1^3 & \ldots & x_3^3 \end{bmatrix}_{\mathbf{nxn}}$$

- Each value in the matrix $X^T X$ is the dot product of values of X^T and X that gives the similarity of data points.

Example, dot product = 0 (no similarity)



no similarity as orthogonal

- Thus it represents pairwise similarities.

Advantages of Kernel PCA

- Can handle non linear data

Disadvantages

- Reconstruction is not possible always for all kernels

tSNE – local embedding

- t-SNE (t-Distributed Stochastic Neighbor Embedding) is an alternative dimensionality reduction algorithm.
- PCA tries to find a global structure
  - Low dimensional subspace
  - Can lead to local inconsistencies
  - Far away point can become nearest neighbors
- t-SNE tries to perserve local structure
  - Low dimensional neighborhood should be the same as original neighborhood.
- Unlike PCA almost only used for visualization
  - No easy way to embed new points

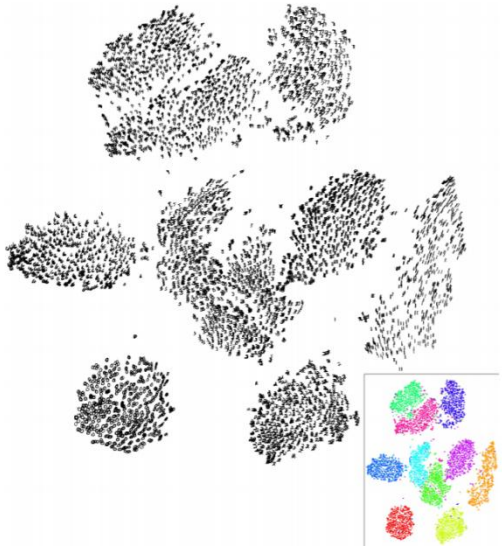
Popular Kernels

Gaussian $K(\vec{x},\vec{x}') = \exp\left(-\beta\|\vec{x}-\vec{x}'\|^2\right)$

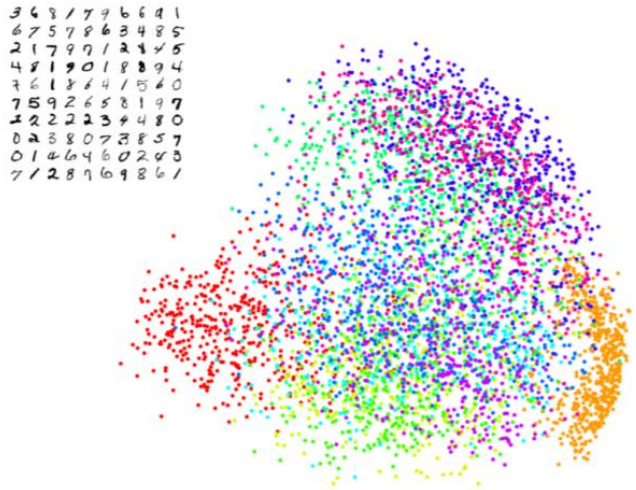
Polynomial $K(\vec{x},\vec{x}') = (1+\vec{x}\cdot\vec{x}')^p$

Hyperbolic tangent $K(\vec{x},\vec{x}') = \tanh(\vec{x}\cdot\vec{x}'+\delta)$

tSNE 2 dimensions embedding for MNIST



PCA 2 dimensions embedding for MNIST



SNE Objective

•Given x (1) , .., x (N) ∈ R D we define the distribution Pij
•Goal: Find good embedding y (1) , .., y (N) ∈ R d for some d < D (normally 2 or 3)
•How do we measure an embedding quality?
•For points y (1) , .., y (N) ∈ R d we can define distribution Q similarly the same (notice no σ 2 i and not symmetric)

$$Q_{ij} = \frac{\exp\left(-||\mathbf{y}^{(i)} - \mathbf{y}^{(j)}||^2\right)}{\sum_k \sum_{l \neq k} \exp\left(-||\mathbf{y}^{(l)} - \mathbf{y}^{(k)}||^2\right)}$$

Optimize Q to be close to P
Minimize KL-divergene
The embeddings y (1) , .., y (N) ∈ R d are the parameters we are optimizing

KL divergence

Measures distance between two distributions, P and Q:

$$KL(Q||P) = \sum_{ij} Q_{ij} \log\left(\frac{Q_{ij}}{P_{ij}}\right)$$

MDS: Multi Dimensional Scaling

- Deal with highly multivariate/ high dimensional datasets
- Projection to a lower dimension may improve interpretability
- Measures of proximity between pairs of objects.
  - ❑ Proximity measure – index over pairs of objects that quantifies the degree to which the two object are alike
  - ❑ Measure of similarity – correspond to stimulus pairs that are alike or close in proximity
  - ❑ Measure of dissimilarity – correspond to stimulus pairs that are least alike or far in proximity
- MDS minimizes the square distance of pairwise distances between all training data with projected, lower dimensional and original feature higher space
- For MDS, stress is minimized such that

$$Stress_p(X_1,..,X_n) = \left(\sum_{i=1}^{n} \sum_{j=1 \mid i \neq j}^{n} (|x_i - x_j| - \delta_{i,j})^2 \right)^{1/2}$$

Where $\delta_{i,j}$ is the general dissimilarity metric in the original dimension and $||x_i - x_j||$ projected lower dimensional dissimilarity between training data *i* and *j*.

Multi Dimensional Scaling Model

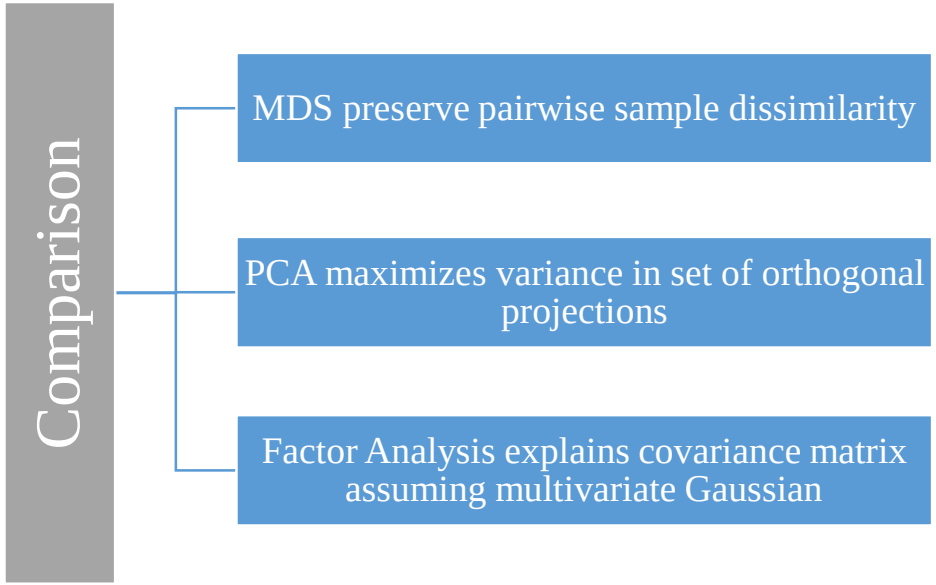
- Classical MDS uses Euclidean principles to model data proximities in geometrical space, where distance (d_{ij}) between *i* and *j* as defined as $d_{ij} = \sqrt{(x_{ia} - x_{ja})^2}$

Xi and Xj specify coordinates of points and j on dimension a, respectively.

The modelled Euclidean distances are related to the proximities δ_{ij}, by some transformation/function (f)

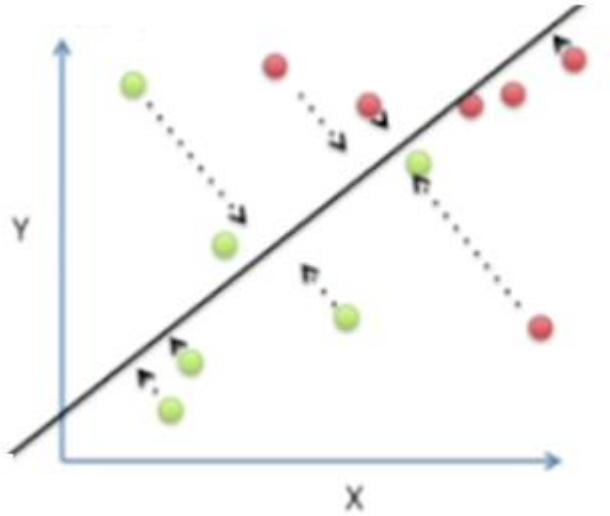
$$d_{ij} = f(d_{ij})\sqrt{(x_{ia} - x_{ja})^2}$$

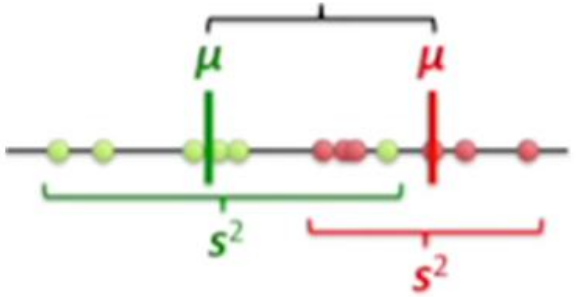
All MDS algorithms are a variation of the above.



Fisher Linear Discriminant Analysis

- The original *dichotomous* discriminant analysis was developed by Sir Ronald Fisher in 1936.
- Discriminant function analysis is useful in determining whether a set of variables is effective in predicting category membership.





$$max \frac{(\mu - \mu)^2}{s^2 + s^2}$$

Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is used to solve dimensionality reduction for data with higher attributes

- Pre-processing step for pattern-classification and machine learning applications.
- Used for feature extraction.
- Linear transformation that maximize the separation between multiple classes.
- "Supervised" - Prediction agent

Feature Subspace :

To reduce the dimensions of a d-dimensional data set by projecting it onto a (k)-dimensional subspace (where k < d)

Feature space data is well represented?

- Compute eigen vectors from dataset
- Collect them in scatter matrix
- Generate *k*-dimensional data from d-dimensional dataset.

Scatter Matrix:

Within class scatter matrix $S_W = \sum_{i=1}^{c} S_i$

Maximize the between class measure & minimize the within class measure

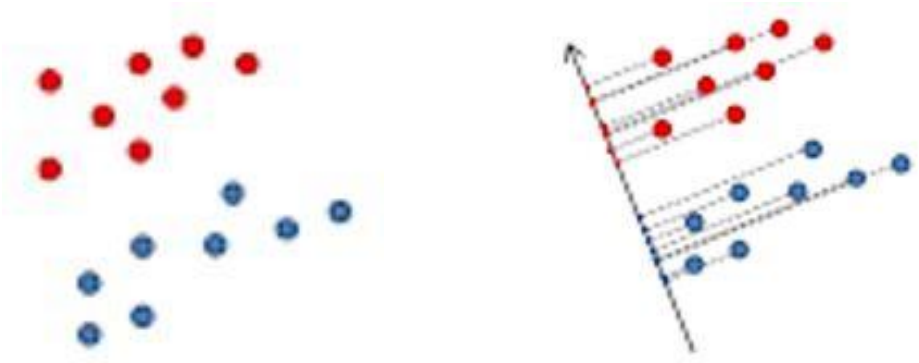
In between class scatter matrix $S_B = \sum_{i=1}^{c} N_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T$

Linear Discriminant Analysis

Between class variance: $\max \left\|w^T \bar{X}_g - w^T \bar{X}\right\|_F^2 = \left\|\bar{Y}_g - \bar{Y}\right\|_F^2$

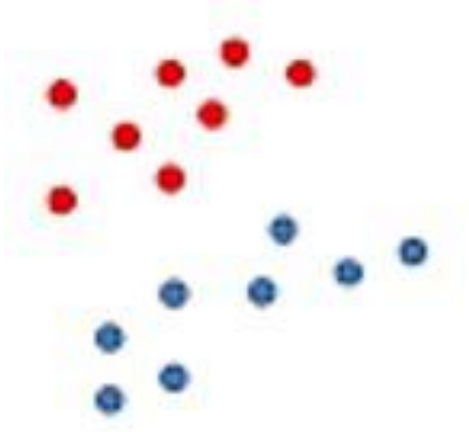
Within class variance: $\min \left\|w^T \bar{X}_{gi} - w^T \bar{X}_g\right\|_F^2$

Fisher Discriminant Ratio: $\max \dfrac{\sum_{g=1}^{C} \left\|w^T \bar{X}_g - w^T \bar{X}\right\|_F^2}{\sum_{g=1}^{C} \sum_{i=1}^{G} \left\|w^T \bar{X}_{gi} - w^T \bar{X}_g\right\|_F^2}$

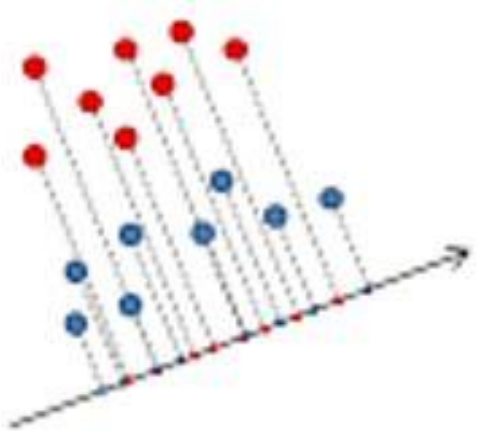


Linear Discriminant Analysis

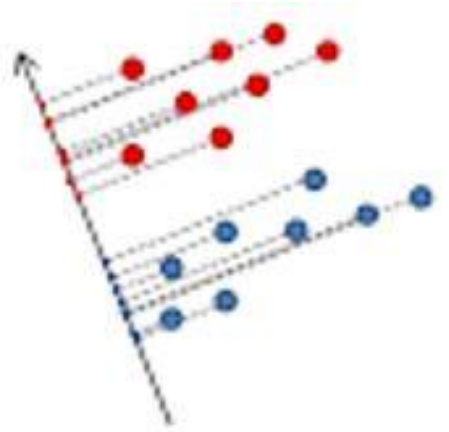
Labelled Data

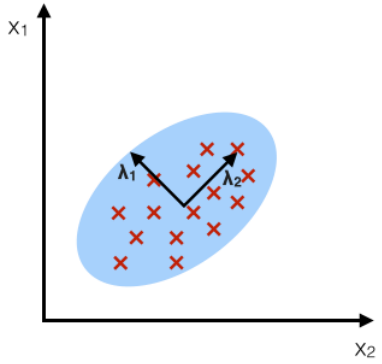


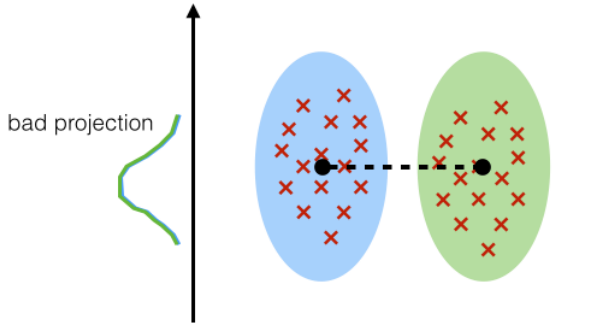
PCA Projection
Maximizing variance of the whole set

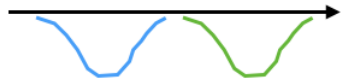


LDA Projection
Maximizing distance between groups









good projection: separates classes well

Linear Discriminant Analysis

General steps to perform LDA:

Step 1: Compute the ***d***-dimensional mean vectors for the different classes from the dataset

Step 2: Compute the scatter matrices S_B and S_W (in-between-class and within-class scatter matrix)

Step 3: Compute the eigenvectors ($\boldsymbol{e_1}, \boldsymbol{e_2}, \dots, \boldsymbol{e_d}$) and corresponding eigenvalues ($\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \dots, \boldsymbol{\lambda_d}$) for the scatter matrices

Step 4: Sort the eigenvectors by decreasing eigenvalues and choose ***k*** eigenvectors with the largest eigenvalues to form a $\boldsymbol{d} \times \boldsymbol{k}$ dimensional matrix ***W*** (where every column represents an eigenvector)

Step 5: Use this $\boldsymbol{d} \times \boldsymbol{k}$ eigenvector matrix to transform the samples onto the new subspace.

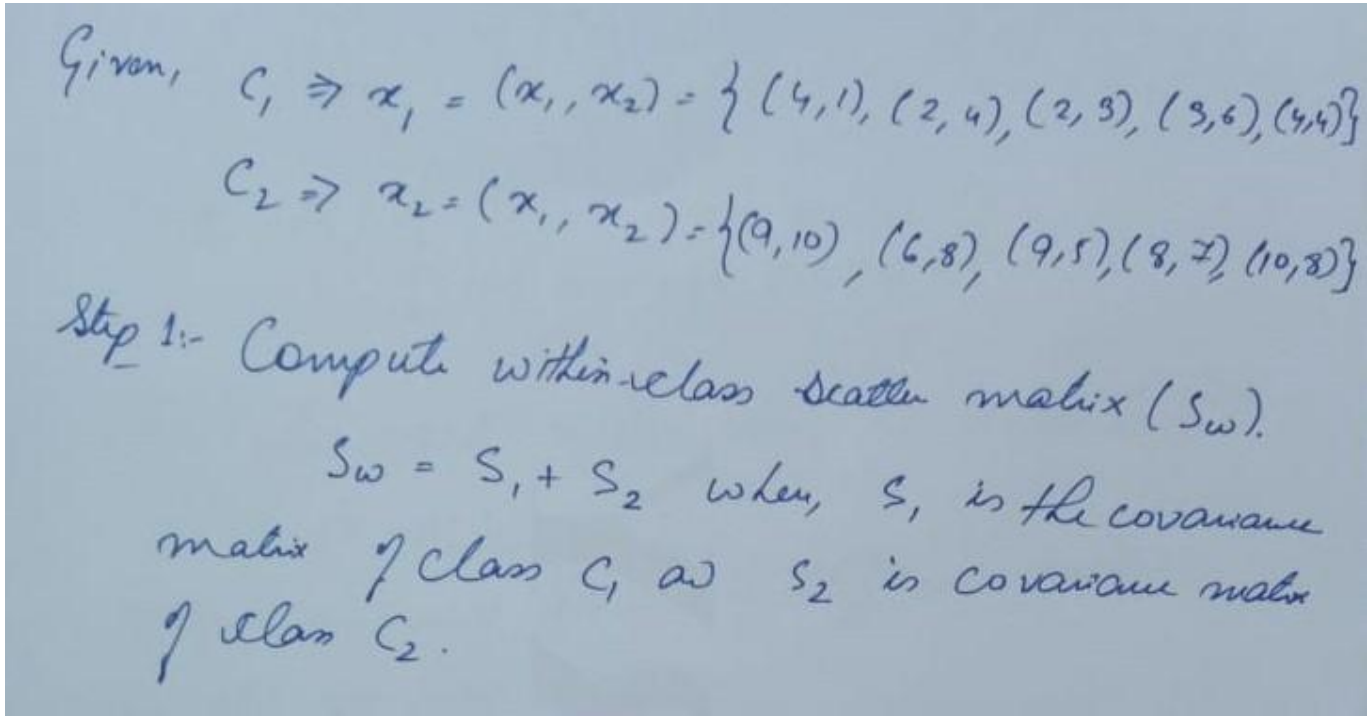
This can be summarized by the matrix multiplication: $\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{W}$ (where ***X*** is a $\boldsymbol{n} \times \boldsymbol{d}$-dimensional matrix representing the ***n*** samples, and ***y*** are the transformed $\boldsymbol{n} \times \boldsymbol{k}$-dimensional samples in the new subspace).

Example

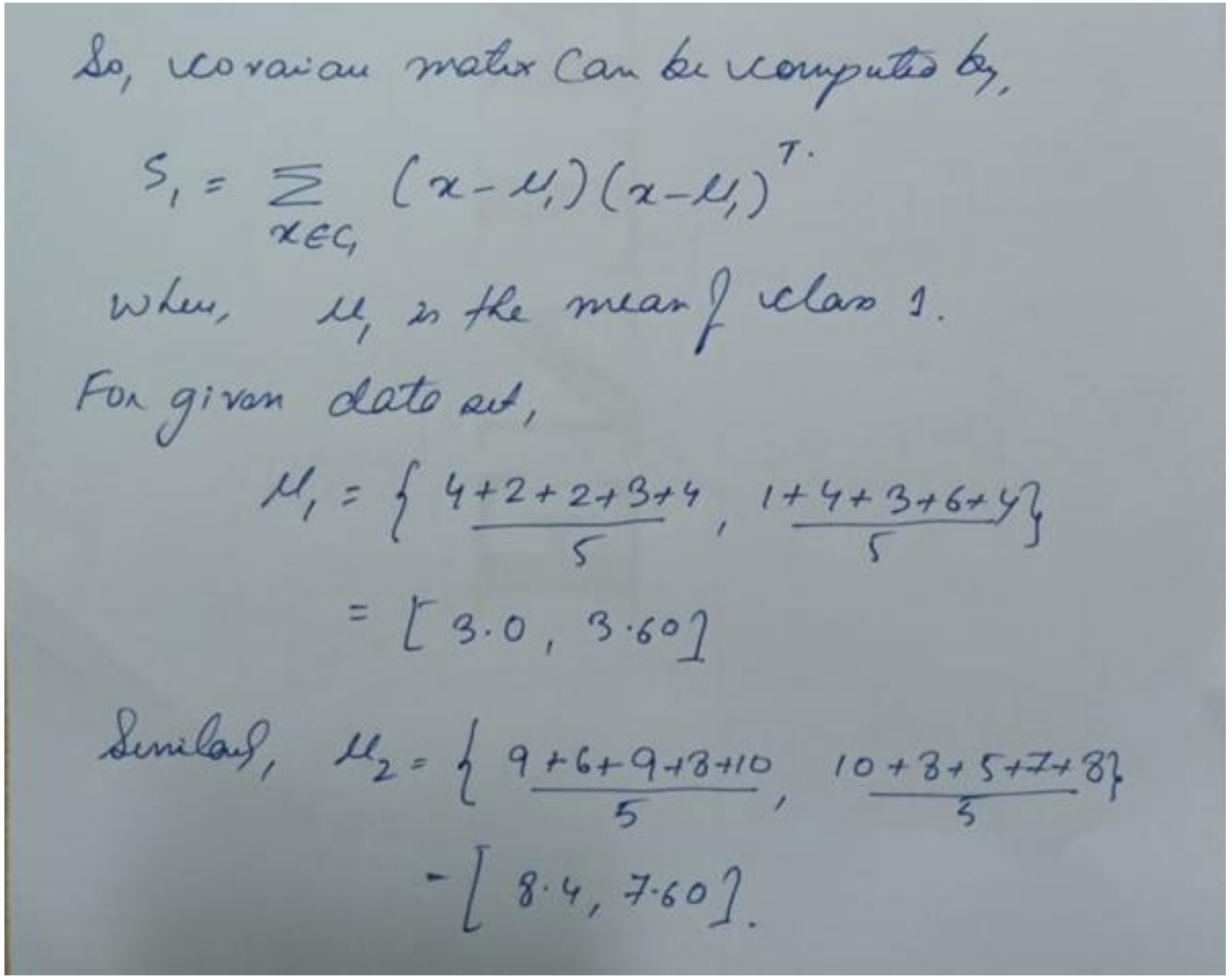
Question: Let us consider the 2D data set given below. Achieve reduced dimension for the dataset using linear discriminant analysis (LDA) technique.

$C_1 \rightarrow X_1 = (x_1, x_2) = \{(4, 1), (2, 4), (2, 3), (3, 6), (4, 4)\}$
$C_2 \rightarrow X_2 = (x_1, x_2) = \{(9, 10), (6, 8), (9, 5), (8, 7), (10, 8)\}$

Solution: Our aim is achieve 2D data points project on a line and convert into 1D



Example



Now, $S_1 = \sum_{x \in \omega_1} (x - \mu_1)(x - \mu_1)^T$, $\mu_1 = [3, 3.6]$

$$\therefore (x - \mu_1) = \begin{bmatrix} 1 & -1 & -1 & 0 & 1 \\ -2.6 & 0.4 & -0.6 & 2.4 & 0.4 \end{bmatrix}$$

Now, for each x, we have to calculate $(x - \mu_1)(x - \mu_1)^T$.

Here, we will have 5 different matrices.

For the 1st value,

$$\begin{bmatrix} 1 \\ -2.6 \end{bmatrix} \begin{bmatrix} 1 & -2.6 \end{bmatrix} = \begin{bmatrix} 1 & -2.6 \\ -2.6 & 6.76 \end{bmatrix} \quad \text{—— (1)}$$

$$\begin{bmatrix} -1 \\ 0.4 \end{bmatrix} \begin{bmatrix} -1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & -0.4 \\ -0.4 & 0.16 \end{bmatrix} \quad \text{—— (2)}$$

$$\begin{bmatrix} -1 \\ -0.6 \end{bmatrix} \begin{bmatrix} -1 & -0.6 \end{bmatrix} = \begin{bmatrix} 1 & 0.6 \\ 0.6 & 0.36 \end{bmatrix} \quad \text{—— (3)}$$

$$\begin{bmatrix} 0 \\ 2.4 \end{bmatrix} \begin{bmatrix} 0 & 2.4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 5.76 \end{bmatrix} \quad \text{—— (4)}$$

$$\begin{bmatrix} 1 \\ 0.4 \end{bmatrix} \begin{bmatrix} 1 & 0.4 \end{bmatrix} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 0.16 \end{bmatrix} \quad \text{—— (5)}$$

Adding ① + ② + ③ + ④ + ⑤ we get

Then once we get the covariance matrix S_1.

$$\therefore \quad S_1 = \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix}$$

Similarly repeat the steps for class 2, (C_2) we can compute the covariance matrix,

$$S_2 = \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}, \quad \mu_2 = [3.4, 7.6]$$

Now, we can compute the within class matrix

$$S_w = S_1 + S_2$$

$$= \begin{bmatrix} 0.8 & -0.4 \\ -0.4 & 2.6 \end{bmatrix} + \begin{bmatrix} 1.84 & -0.04 \\ -0.04 & 2.64 \end{bmatrix}$$

$$= \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}$$

Step 2:- We have to now calculate between class scatter matrix, which is given by.

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

So, we have, $\mu_1 = [3.0 \quad 3.6]$

$\mu_2 = [8.4, \quad 7.6]$.

$\therefore (\mu_1 - \mu_2) = [-5.4 \quad -4]$.

$\therefore S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$

$= \begin{bmatrix} -5.4 \\ -4 \end{bmatrix} [-5.4 \quad -4]$

$= \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16.0 \end{bmatrix}$

Step 3:- Find the best LDA projection vector. This can be found using eigen vector having largest eigen value (like PCA) or alternatively by using inverse matrix.

Here we will compute using inverse matrix.

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = S_w^{-1} (\mu_1 - \mu_2).$$

* Inverse matrix can be computed by,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

So, in our case we have,

$$S_w = \begin{bmatrix} 2.64 & -0.44 \\ -0.44 & 5.28 \end{bmatrix}$$

$$\therefore S_w^{-1} = \frac{1}{13.74} \begin{bmatrix} 5.28 & 0.44 \\ 0.44 & 2.64 \end{bmatrix}$$

$$= \begin{bmatrix} 0.384 & 0.032 \\ 0.032 & 0.192 \end{bmatrix}$$

$$\therefore \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = S_w^{-1} (\mu_1 - \mu_2)$$

$$= \begin{bmatrix} 0.384 & 0.032 \\ 0.032 & 0.192 \end{bmatrix} \begin{bmatrix} -5.4 \\ -4 \end{bmatrix}$$

$$= \begin{bmatrix} -2.204 & -2.496 \end{bmatrix}^T$$

$$= \begin{bmatrix} -2.204 \\ -2.496 \end{bmatrix}$$

$$\therefore v_1 = -2.204, \quad v_2 = -2.496.$$

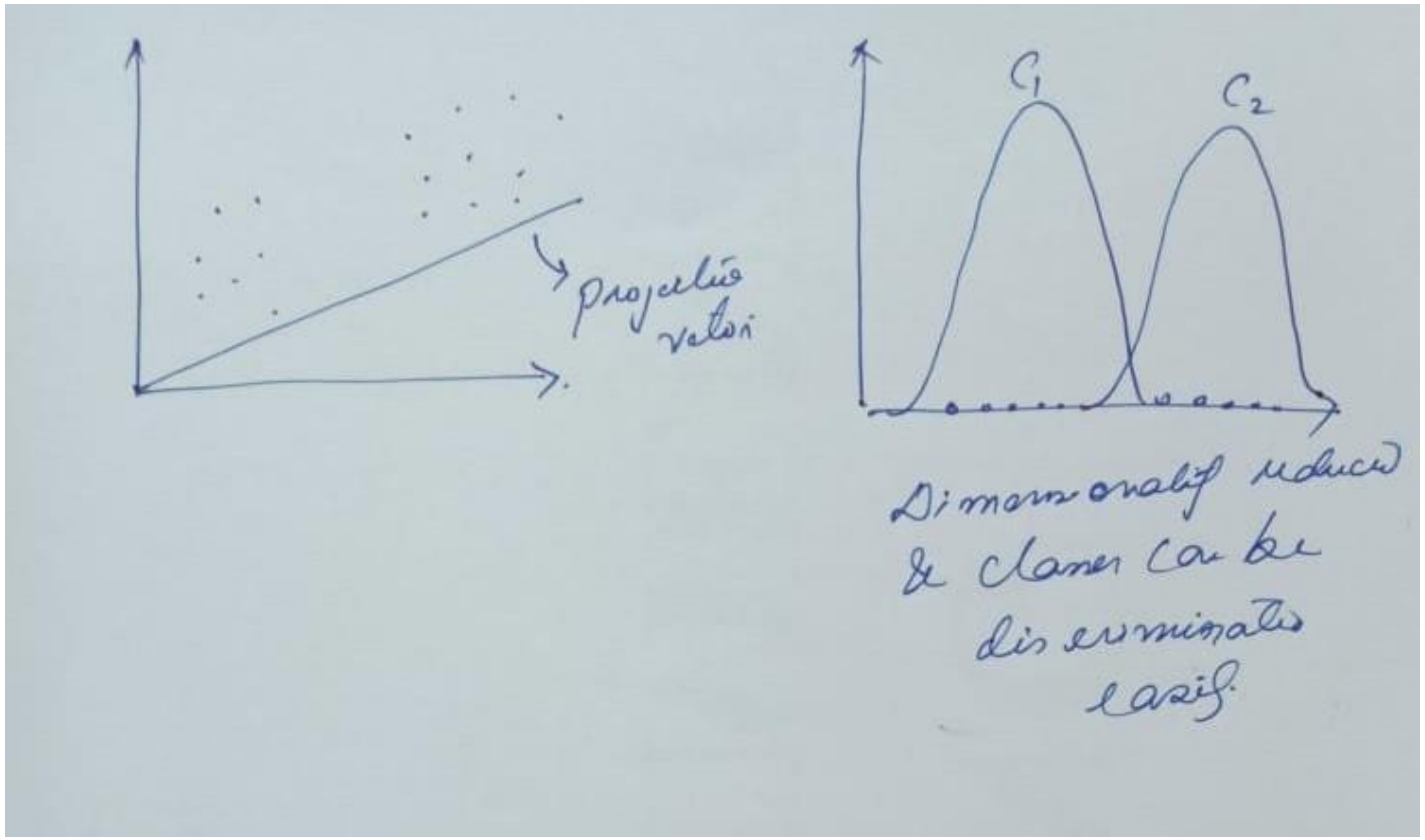
Step 4:- Dimension reduction.

$$Y = w^T X \Rightarrow \text{input data sample.}$$

$\Downarrow$

Projection value

Example



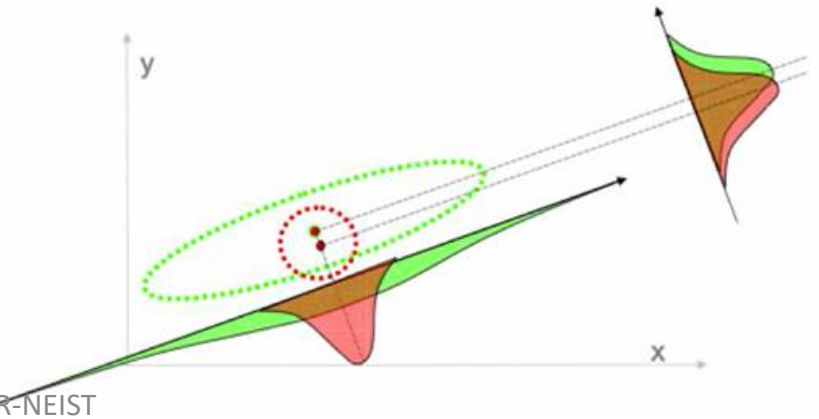
Is LDA always better than PCA?

- There has been a tendency in the computer vision community to prefer LDA over PCA.
- This is mainly because LDA deals directly with discrimination between classes while PCA does not pay attention to the underlying class structure.

LDA always guaranteed to be better for classification over PCA

- LDA assumes classes are unimodal Gaussians
- Fails when discriminatory information is not in the same mean, but in the variance of the data

Counter example *(PCA performs better projection)*



Gaussian Discriminant Analysis

Learning algorithm can be broadly of two types:

- **Discriminative:** These types of algorithm tries to find a decision boundary between different classes during learning process.

Learns p (y | x) i.e. given a feature set X for a data sample what is the probability it belongs to the class 'y'

or $h(x) = \begin{cases} 0 \\ 1 \end{cases}$ on some condition that $x \rightarrow y$

- **Generative:** These algorithms try to capture the distribution of each class separately instead of finding a decision boundary among classes

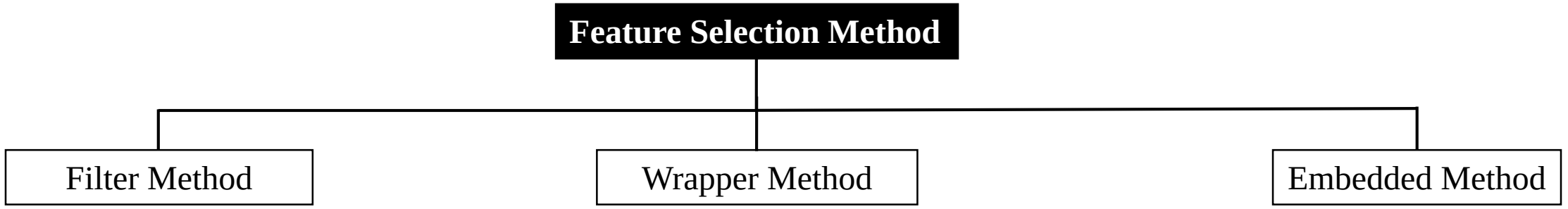
Learns p (x | y) i.e. given a class, the learning algorithm can identify the features with p(y) being the class prior

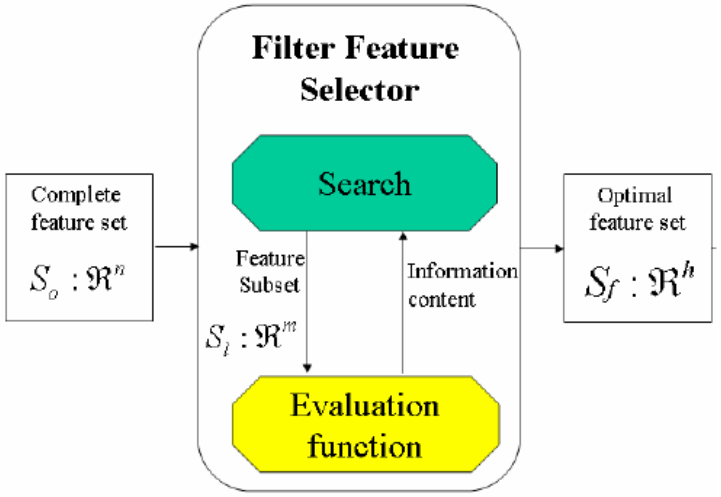
The prediction in generative learning is done using Baye's algorithm

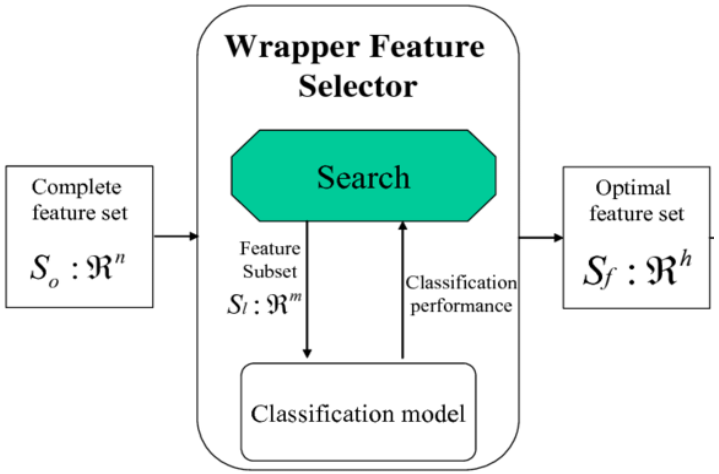
$P(y|X) = \frac{P(X|y).P(y)}{P(X)}$ where $P(X) = P(X|y = 1).P(y = 1) + P(X|y = 0).P(y = 0)$

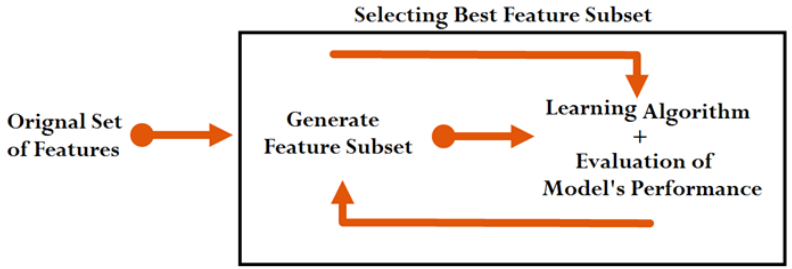
Gaussian Discriminant Analysis (GDA) is a generative learning algorithm which tries to fit a Gaussian distribution to every class separately

Supervised Reduction Techniques

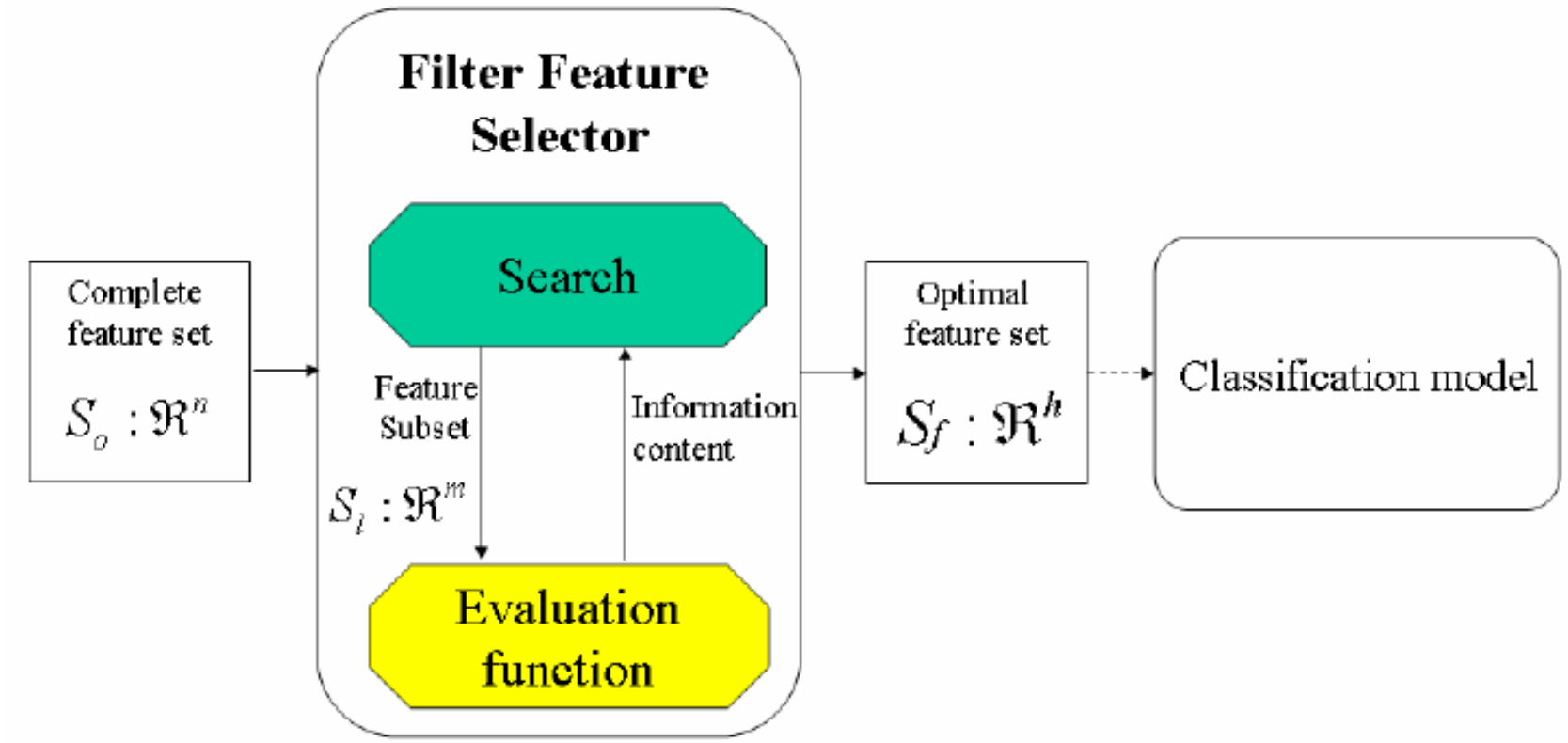








Filter Method



Filter Method

Search Strategy

Statistical based Methods

- Low Variance (Variance Threshold)
- Pearson Correlation Coefficient Score
- T-Score (binary classification)
- Chi-Square Score
- ANOVA

Cons: Cannot Handle Feature Redundancy

Search Strategy

Information Theoretical Methods

- Information Gain (Maximum Correlation with class label)
- Minimum Redundancy Maximum Relevance (MRMR)
  - Maximum Correlation with class label
  - Less dependent or not correlated with each other
- Conditional Information Feature Selection (CIFS)
  - Maximum Correlation with class label
  - Less dependent or not correlated with each other
  - Takes care of redundancy between features that are not selected and selected

Cons: Cannot Handle Unsupervised Scenario and Cannot Handle continuous data

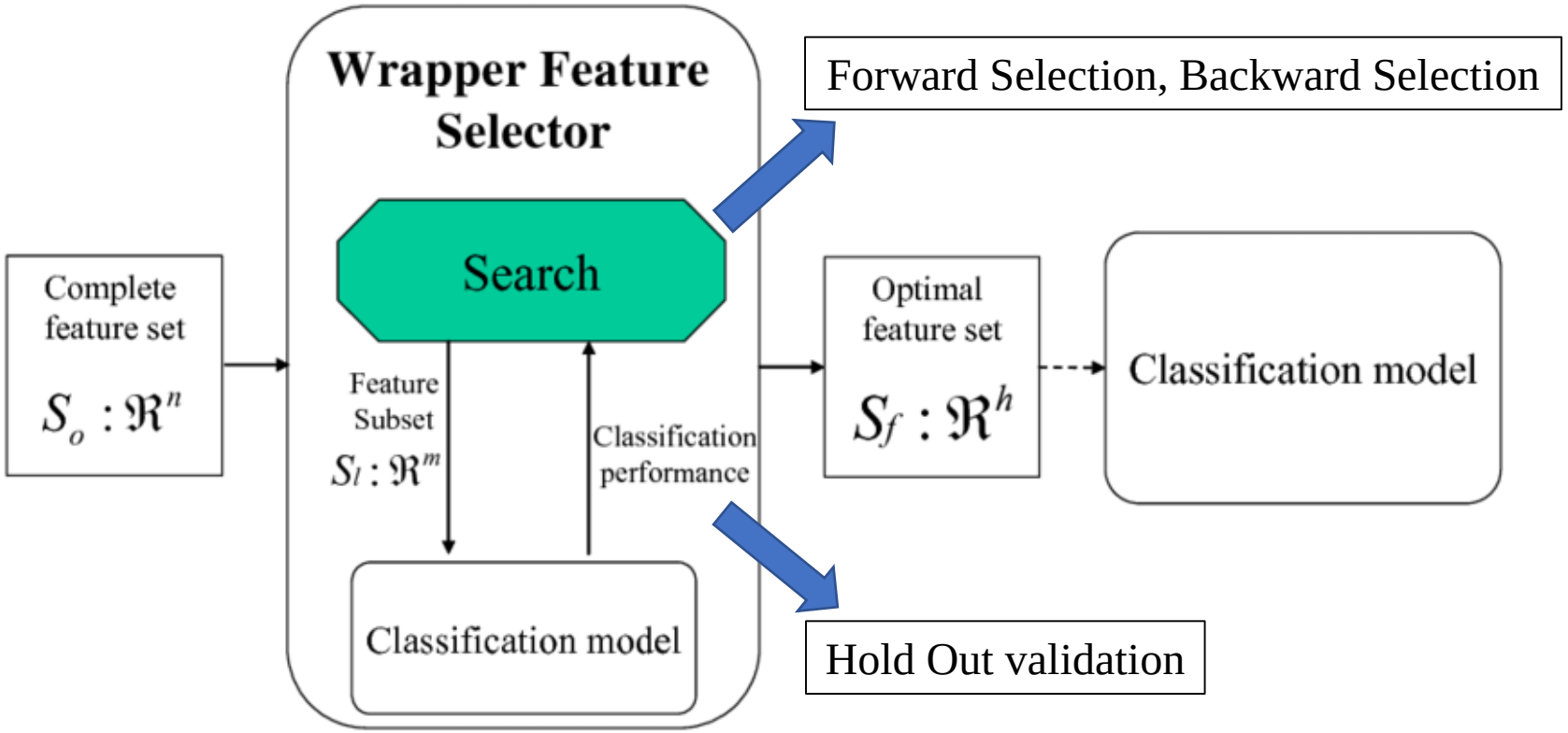
Search Strategy

Similarity Based Methods (Preserve Data Similarity)

- Similarity based on labels (Supervised)
- Similarity based on distance metric (Unsupervised)
- Affinity Matrix (pairwise similarity matrix)
- Laplacian Score (Unsupervised)
- SPEC (Supervised + Unsupervised)
- Fisher Score (Supervised)
- ReleifF (Supervised)

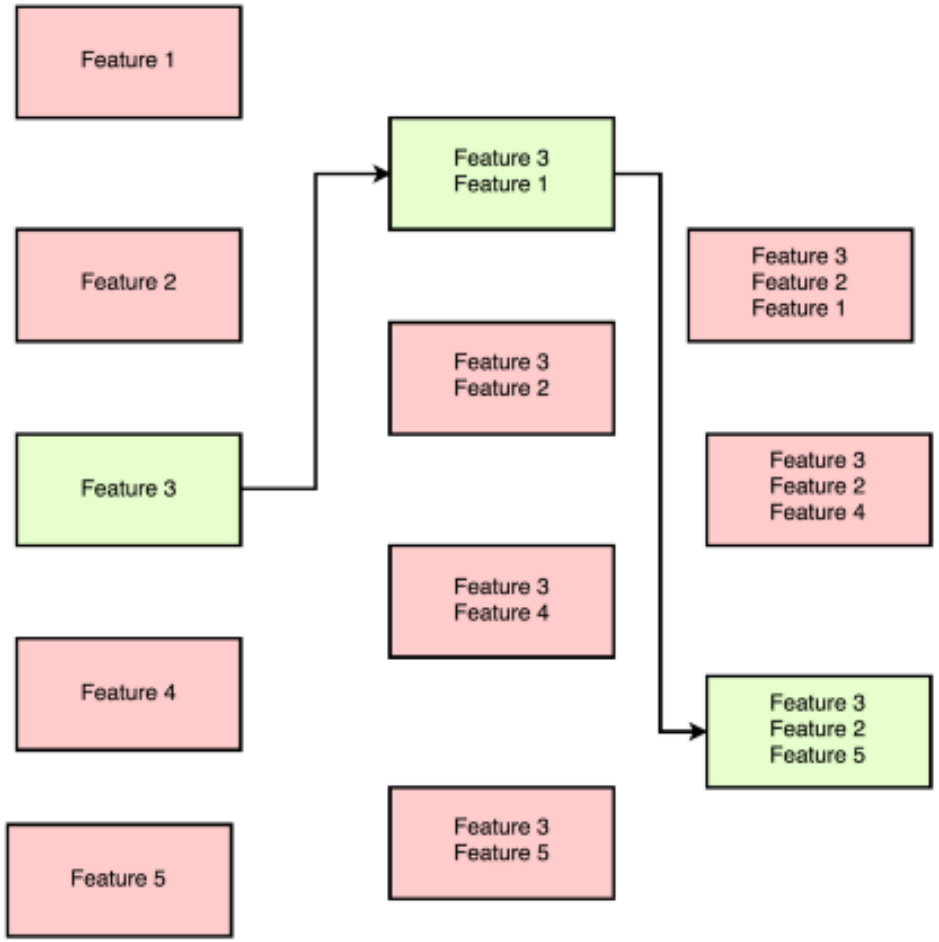
Cons: Most Cannot Handle Feature Redundancy

Wrapper Method

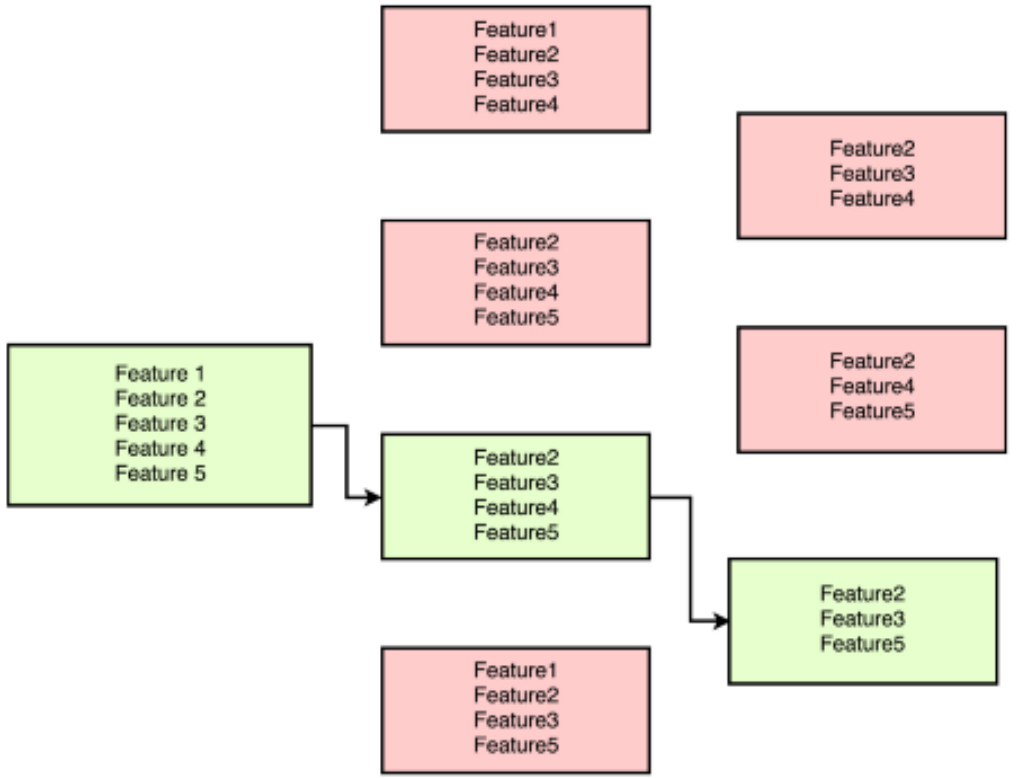


Wrapper Method

Search Strategy

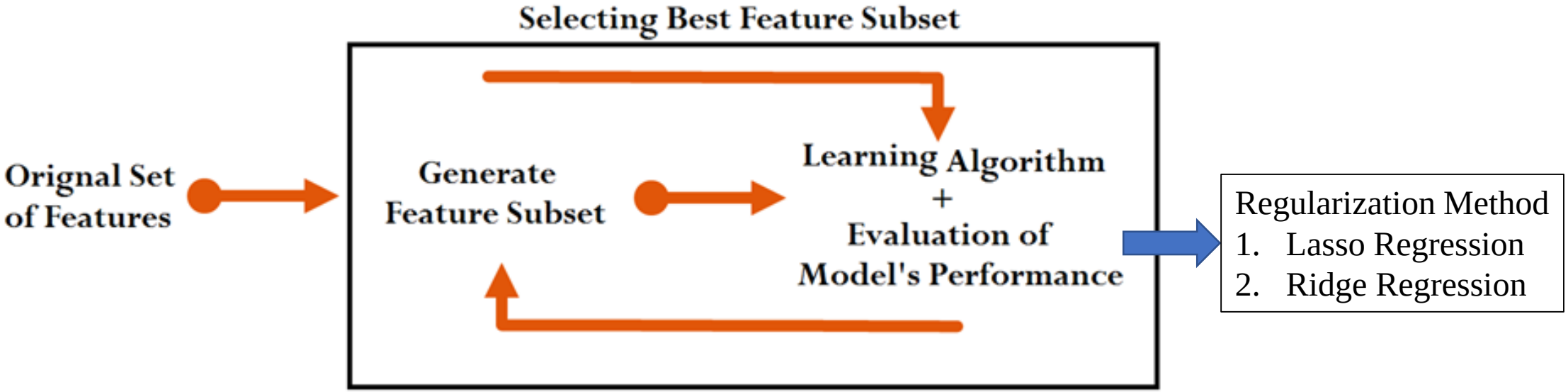


Forward Selection



Backward Selection

Embedded Method



$$\min_{w} (\hat{y} - Y)2 + ||w||1$$

$$\min_{w} (\hat{y} - w_0 + W_1X_1 + W_2X_2 + \ldots + WdXd_{)2} + ||w||1$$

Advantages of Feature Selection Methods

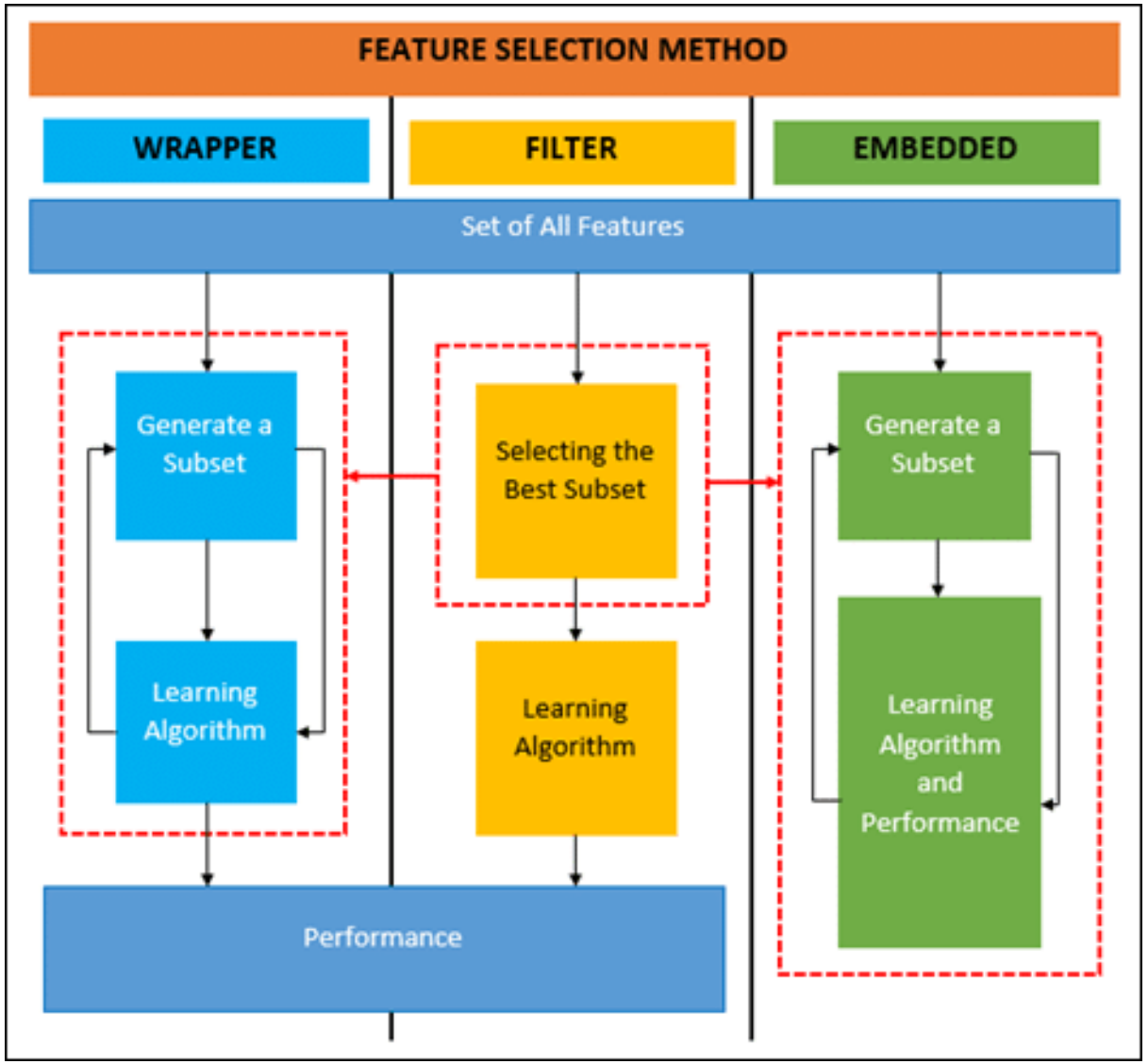
❑ **Improve Model Performance:**

- ✓ Alleviate curse of dimensionality
- ✓ Takes care of multicollinearity
- ✓ Reduce space and time complexity

❑ **Model Interpretation (w.r.t. to features):** As reduced Features are kept same as original Features

❑ **Improving Data Acquisition Process:** As reduced Features are kept same as original Features

Feature Selection In Nutshell



Worked Examples

Q. 1 From the following data; use PCA to reduce the dimension from 2 × 4 to 1 × 4.

Qu- For the following data use PCA to reduce the dim. from 2 to 1 :

feature	ex_1	ex_2	ex_3	ex_4
x	4	8	13	7
y	11	4	5	14

Data set-

Step ① no. of features = n = 2

no. of samples = N = 4

Step ② Mean of variables-

$$\bar{x} = \frac{4+8+13+7}{4} = 8$$

$$\bar{y} = \frac{11+4+5+14}{4} = 8.5$$

Step③ Computation of Covariance Matrix –

The ordered pairs are – (x,x) (x,y) (y,x) (y,y)

- Covariance of the ordered pair (x_i, x_j) :–

$$\text{Cov}(x_i, x_j) = \frac{1}{N-1} \sum_{k=1}^{N} (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)$$

@ $x_i = x_j \Rightarrow$

$$\text{Cov}(x,x) = \frac{1}{N-1} \sum_{k=1}^{N} (x - \bar{x})^2$$

Now here, $\text{Cov}(x,x) = \frac{1}{4-1}\{(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2\}$

$= 14$

$\text{Cov}(x,y) = \frac{1}{4-1}\{(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)\}$

$= -11$

$\because \text{cov}(x,y) = \text{cov}(y,x) = -11$

Now, $\text{cov}(y,y) = \frac{1}{4-1}\{(11-8.5)^2 + (4-8.5)^2 + (5-8.5)^2 + (14-8.5)^2\}$

$= 23$

$\therefore$ the covariance matrix is $(S) = \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$.

Step ④ Eigen value, eigen vector & Normalized Eigen Vector

- Eigen Value – $|S - \lambda I| = 0$

$\Rightarrow \begin{vmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{vmatrix} = 0$

$\Rightarrow (14-\lambda)(23-\lambda) - (-11)^2 = 0$

$\Rightarrow \lambda^2 - 37\lambda + 201 = 0$

for, $ax^2 + bx + c = 0$

$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$

$\lambda = 30.3849, 6.6151$

Now, $\lambda_1 > \lambda_2$

$\Rightarrow \lambda_1 = 30.3849$ (PC_1)

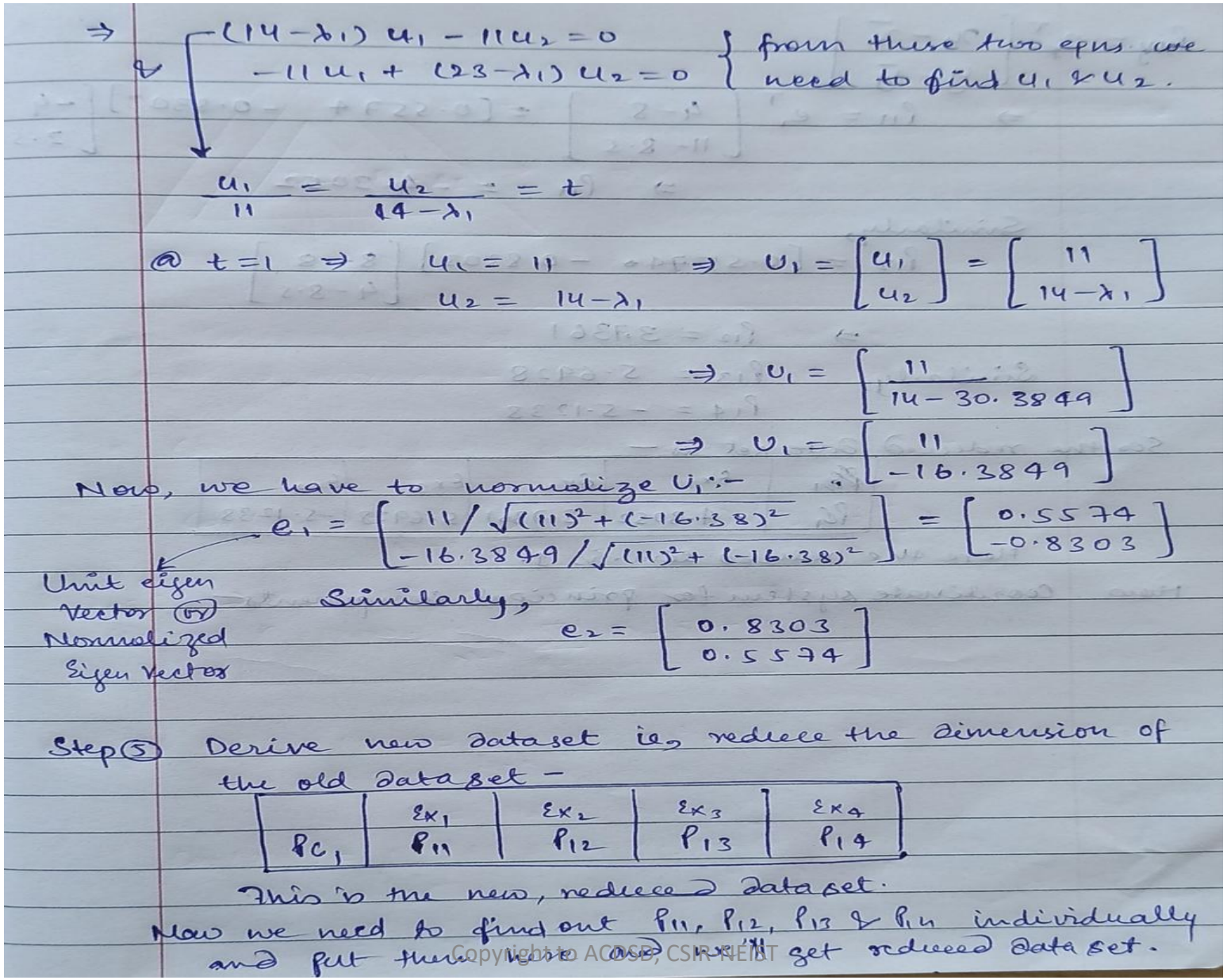
$\lambda_2 = 6.6151$ (PC_2)

Now, Eigen vector of λ_1 :–

$(S - \lambda_1 I) U_1 = 0$

$\begin{bmatrix} 14-\lambda_1 & -11 \\ -11 & 23-\lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$

$\Rightarrow \begin{bmatrix} (14-\lambda_1)u_1 - 11u_2 \\ -11u_2 + (23-\lambda_1)u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$



$\Rightarrow$

$$\begin{cases} (14-\lambda_1)u_1 - 11u_2 = 0 \\ -11u_1 + (23-\lambda_1)u_2 = 0 \end{cases}$$

from these two eqns we need to find u_1 & u_2.

$$\frac{u_1}{11} = \frac{u_2}{14-\lambda_1} = t$$

@ $t=1 \Rightarrow$ $u_1 = 11$, $u_2 = 14-\lambda_1$

$$\Rightarrow U_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 11 \\ 14-\lambda_1 \end{bmatrix}$$

$$\Rightarrow U_1 = \begin{bmatrix} 11 \\ 14 - 30.3849 \end{bmatrix}$$

$$\Rightarrow U_1 = \begin{bmatrix} 11 \\ -16.3849 \end{bmatrix}$$

Now, we have to normalize U_1:-

$$e_1 = \begin{bmatrix} 11/\sqrt{(11)^2 + (-16.38)^2} \\ -16.3849/\sqrt{(11)^2 + (-16.38)^2} \end{bmatrix} = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

Unit eigen vector (or) Normalized Eigen vector

Similarly,

$$e_2 = \begin{bmatrix} 0.8303 \\ 0.5574 \end{bmatrix}$$

Step (5) Derive new dataset ie, reduce the dimension of the old dataset -

	Ex_1	Ex_2	Ex_3	Ex_4
PC_1	P_{11}	P_{12}	P_{13}	P_{14}

This is the new, reduced dataset.

Now we need to find out P_{11}, P_{12}, P_{13} & P_{14} individually and put them into (5) to get reduced dataset.

$P_{xy} = e_k^T \begin{bmatrix} x - \bar{x} \\ y - \bar{y} \end{bmatrix}$

$\Rightarrow \quad P_{11} = e_1^T \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix} = [0.5574 \quad -0.8303] \begin{bmatrix} -4 \\ 2.5 \end{bmatrix}$

$\Rightarrow \quad P_{11} = -4.3052$

Similarly,

$P_{12} = [0.5574 \quad -0.8303] \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix}$

$\Rightarrow \quad P_{12} = 3.7361$

Similarly, $P_{13} = 5.6928$

$P_{14} = -5.1238$

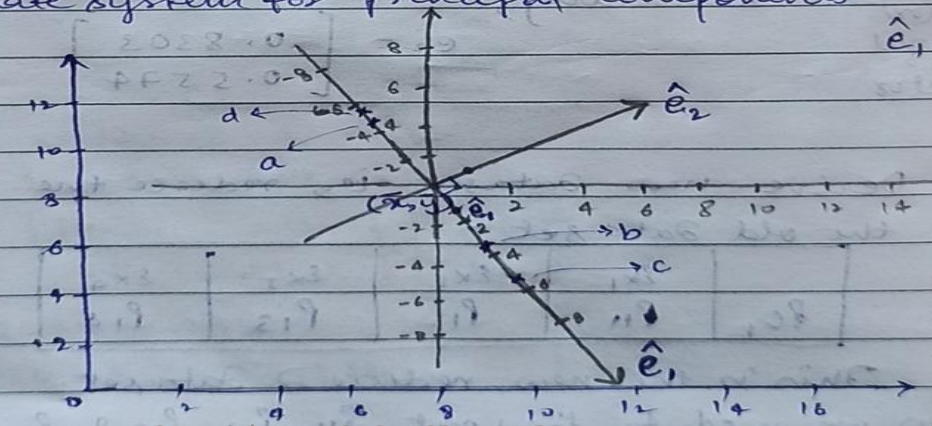
So the reduced data set is —

	e_{x1}	e_{x2}	e_{x3}	e_{x4}
P_{C_1}	ⓐ -4.3052	ⓑ 3.7361	ⓒ 5.6928	ⓓ -5.1238

Here the reduced dim. is 1.

Now, Coordinate system for principal components —

$\hat{e}_1 \perp \hat{e}_2$



Worked Examples

Q. 2 From the following data; construct covariance matrix:

Variable	Ex_1	Ex_2	Ex_3	Ex_4
A	10	16	13	16
B	11	15	12	14

Solution:

$n = 2$

$N = 4$

$$\bar{A} = \frac{10+16+13+16}{4} = \frac{55}{4} = 13.8$$

$$\bar{B} = \frac{11+15+12+14}{4} = \frac{52}{4} = 13$$

Now ord. pairs are —

(x,x), (x,y) (y,x) (x,y)

$$\begin{bmatrix} (x,x) & (x,y) \\ (y,x) & (y,y) \end{bmatrix}$$

Now,

$$Cov(x,x) = \frac{1}{4-1}\left\{(13.8-10)^2 + (13.8-16)^2 + (13.8-13)^2 + (13.8-16)^2\right\}$$

$$= \frac{1}{3}\left\{14.44 + 4.84 + 0.64 + 4.84\right\} = \frac{1}{3}\left\{24.76\right\} = 8.25$$

$$Cov(x,y) = Cov(y,x)$$

$$= \frac{1}{4-1}\left\{(13.8-10)(13-11) + (13.8-16)(13-15) + (13.8-13)(13-12) + (13.8-16)(13-14)\right\}$$

$$= \frac{1}{3}\left\{7.6 + 4.4 + 0.8 - 2.2\right\}$$

$$= \frac{1}{3}\left\{10.6\right\} = 3.53$$

Worked Examples

$$Cov(y,y) = \frac{1}{4-1}\left\{(13-11)^2 + (13-15)^2 + (13-12)^2 + (13-14)^2\right\}$$

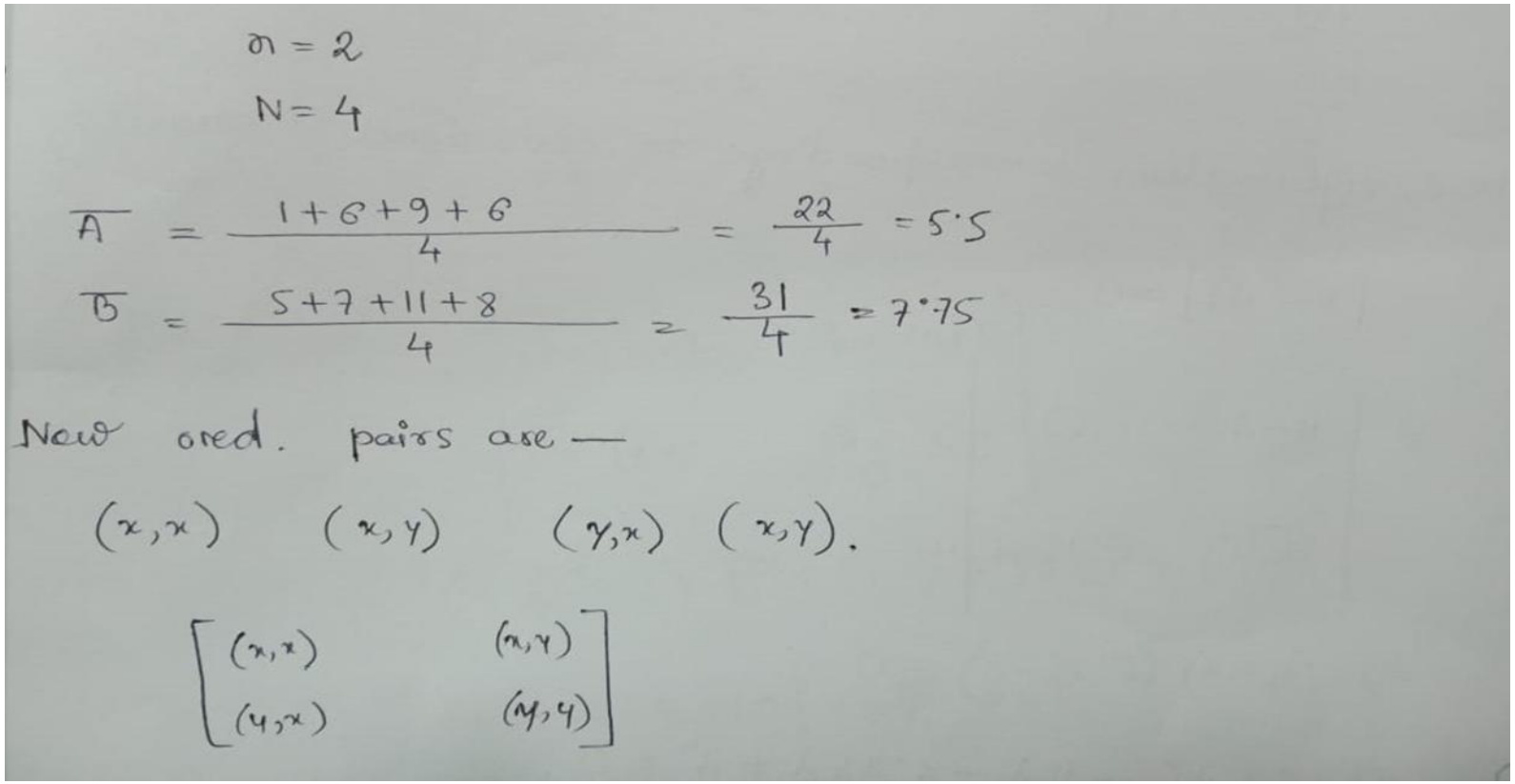
$$= \frac{1}{3}\left\{4+4+1+1\right\} = \frac{10}{3} = 3.33$$

$$\therefore \text{Cov. Mat.} = \begin{bmatrix} 8.25 & 3.53 \\ 3.53 & 3.33 \end{bmatrix}$$

Q 3: From the following data; construct covariance matrix & find eigen values:

Solution:

Variable	Ex_1	Ex_2	Ex_3	Ex_4
A	1	6	9	6
b	5	7	11	8



Now,

$$Cov(x,x) = \frac{1}{4-1}\left\{(1-5\cdot5)^2 + (6-5\cdot5)^2 + (9-5\cdot5)^2 + (6-5\cdot5)^2\right\}$$

$$= \frac{1}{3}\left\{20\cdot25 + 0\cdot25 + 12\cdot25 + 0\cdot25\right\} = 11$$

$$Cov(x,y) = Cov(y,x)$$

$$= \frac{1}{4-1}\left\{(1-5\cdot5)(5-7\cdot75) + (6-5\cdot5)(7-7\cdot75) + (9-5\cdot5)(11-7\cdot75) + (6-5\cdot5)(8-7\cdot75)\right\}$$

$$= \frac{1}{3}\left\{(-2\cdot75) + (-0\cdot75) + (3\cdot25) + (0\cdot25)\right\} = 0$$

$$Cov(Y,Y) = \frac{1}{4-1}\left\{(5-7.75)^2 + (7-7.75)^2 + (11-7.75)^2 + (8-7.75)^2\right\}$$

$$= \frac{1}{3}\left\{7.5625 + 0.5625 + 10.5625 + 0.0625\right\}$$

$$\approx 6.25.$$

$$\therefore \text{Cov. Mat}_{(S)} = \begin{bmatrix} 11 & 0 \\ 0 & 6.25 \end{bmatrix}$$

Now, eigen values corresponding to Covariance Matrix

$$|S - \lambda I| = 0$$

$$\Rightarrow \begin{vmatrix} 11-\lambda & 0 \\ 0 & 6.25-\lambda \end{vmatrix} = 0$$

$$\Rightarrow (11-\lambda)(6.25-\lambda) = 0$$

$$\Rightarrow 68.75 - 11\lambda - 6.25\lambda + \lambda^2 = 0.$$

$$\Rightarrow \lambda^2 - 17.25\lambda + 68.75 = 0.$$

$a = 1$, $b = -17.25$, $c = 68.75$.

$$\lambda = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= \frac{17.25 \pm \sqrt{(-17.25)^2 - (4 \times 1 \times 68.75)}}{2}$$

$$= \frac{17.25 \pm 4.75}{2}$$

$\lambda_1 = 11$ $\lambda_2 = 6.25$ Real Eigen values.

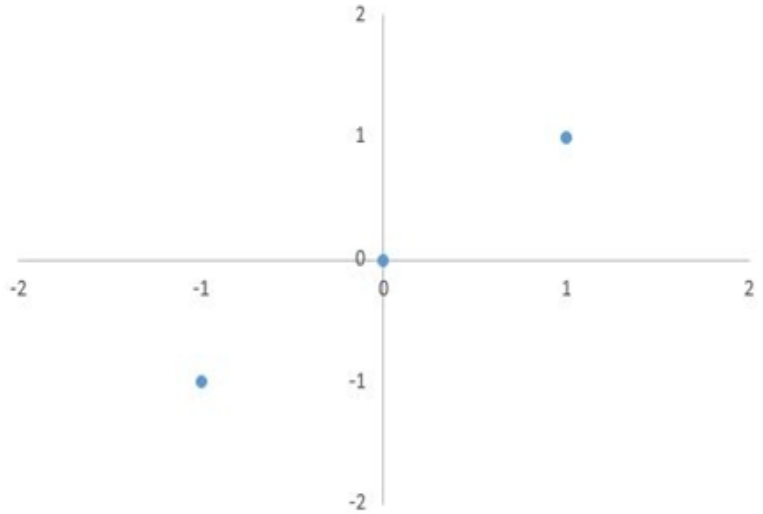
Worked Examples

Q. 4: Consider 3 data points in the 2-d space: (-1, -1), (0, 0) & (1, 1)

What will be the principal component of these data.

1) [√2/2, √2/2] 2) [1/√3, 1/√3]

3) [-√2/2, -√2/2] 4) [-1/√3, -1/√3]

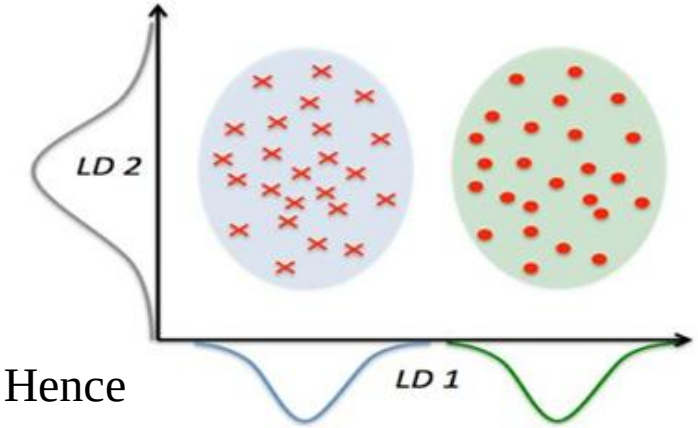


a) 1 and 2
b) 3 and 4
c) 1 and 3
d) 2 and 4

Solution: The correct answer is c) because the principal components should be normalized to have unit length.

Worked Examples

Q. 5: In LDA, the idea is to find the line that best separates the two classes. In the given image which of the following is a good projection?



a) LD1
b) LD2
c) Both
d) None of these

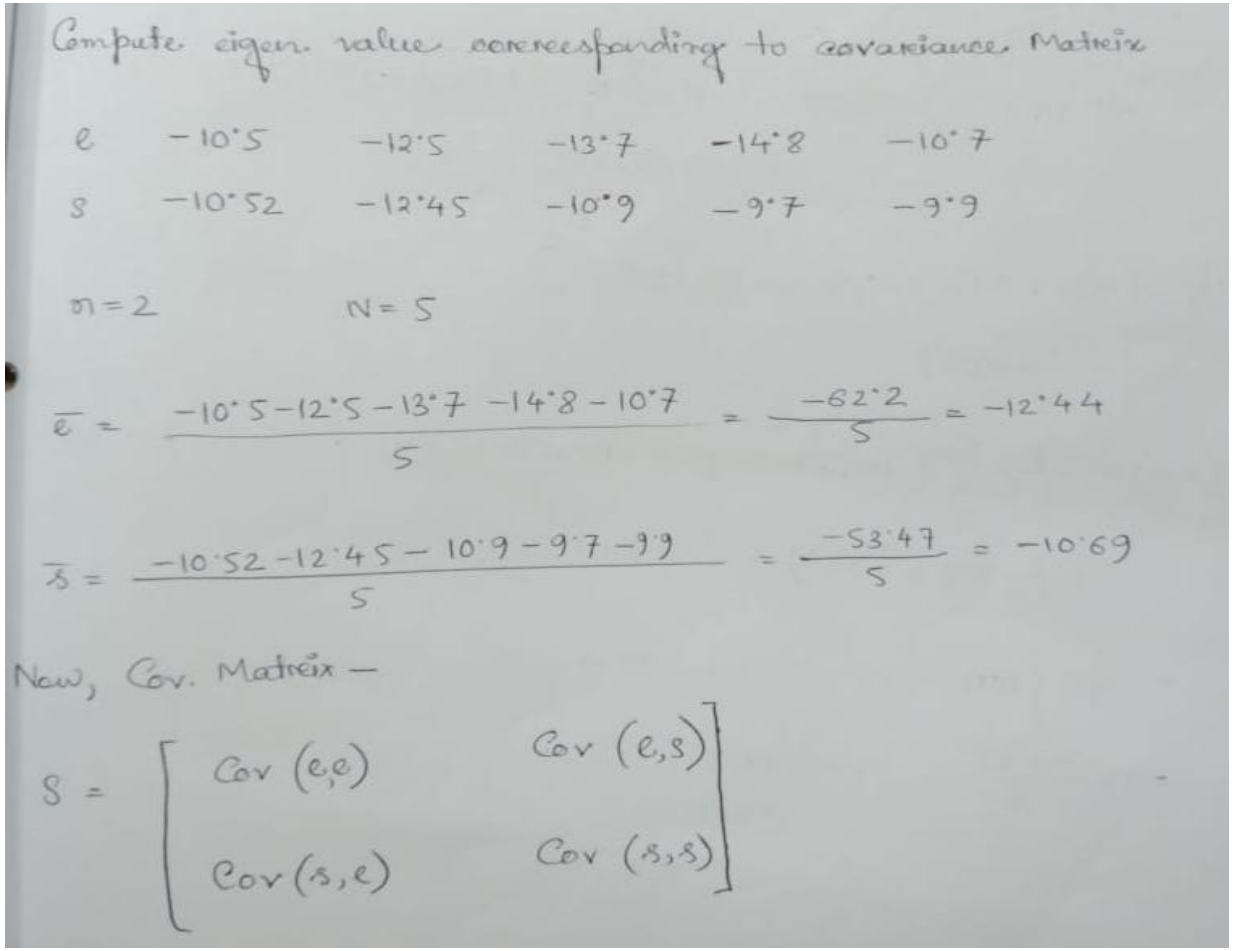
Solution: Clearly, LD1 is a good projection as it separates both the classes. Hence the correct choice is a).

Worked Examples

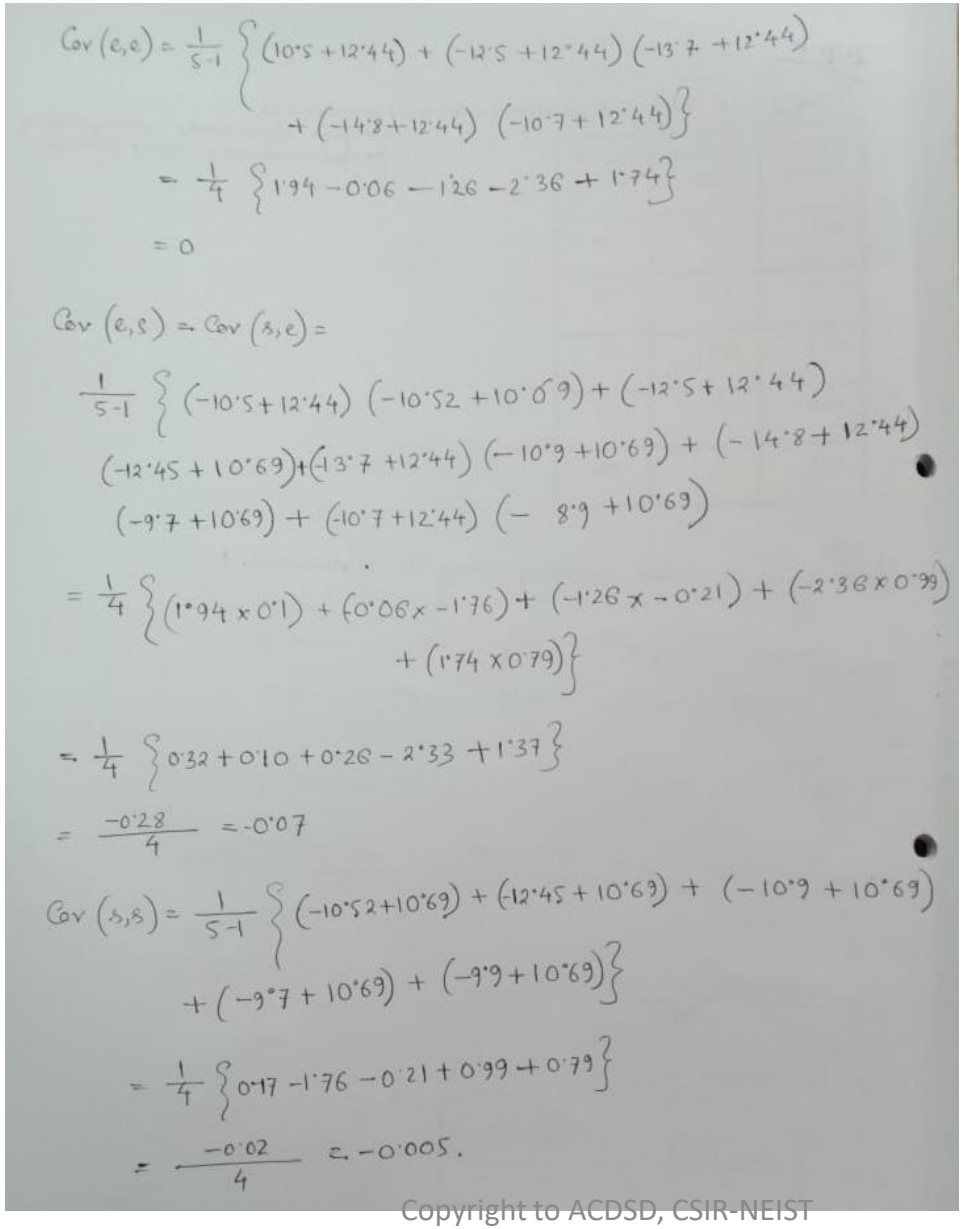
Q. 6 A docking data is summarized as follows:
Compute the Eigen values corresponding to the covariance matrix formed.

Docking Energy	Docking Score	Remark
-10.5	-10.52	Yes
-12.5	-12.45	Yes
-13.7	-10.9	No
-14.8	-9.7	No
-10.7	-9.9	No

Solution:



Worked Examples



$$Cov(e,e) = \frac{1}{5-1}\{(10.5+12.44) + (-12.5+12.44)(-13.7+12.44) + (-14.8+12.44)(-10.7+12.44)\}$$

$$= \frac{1}{4}\{1.94 - 0.06 - 1.26 - 2.36 + 1.74\}$$

$$= 0$$

$$Cov(e,s) = Cov(s,e) =$$

$$\frac{1}{5-1}\{(-10.5+12.44)(-10.52+10.69) + (-12.5+12.44)(-12.45+10.69) + (-13.7+12.44)(-10.9+10.69) + (-14.8+12.44)(-9.7+10.69) + (-10.7+12.44)(-8.9+10.69)$$

$$= \frac{1}{4}\{(1.94 \times 0.1) + (0.06 \times -1.76) + (-1.26 \times -0.21) + (-2.36 \times 0.99) + (1.74 \times 0.79)\}$$

$$= \frac{1}{4}\{0.32 + 0.10 + 0.26 - 2.33 + 1.37\}$$

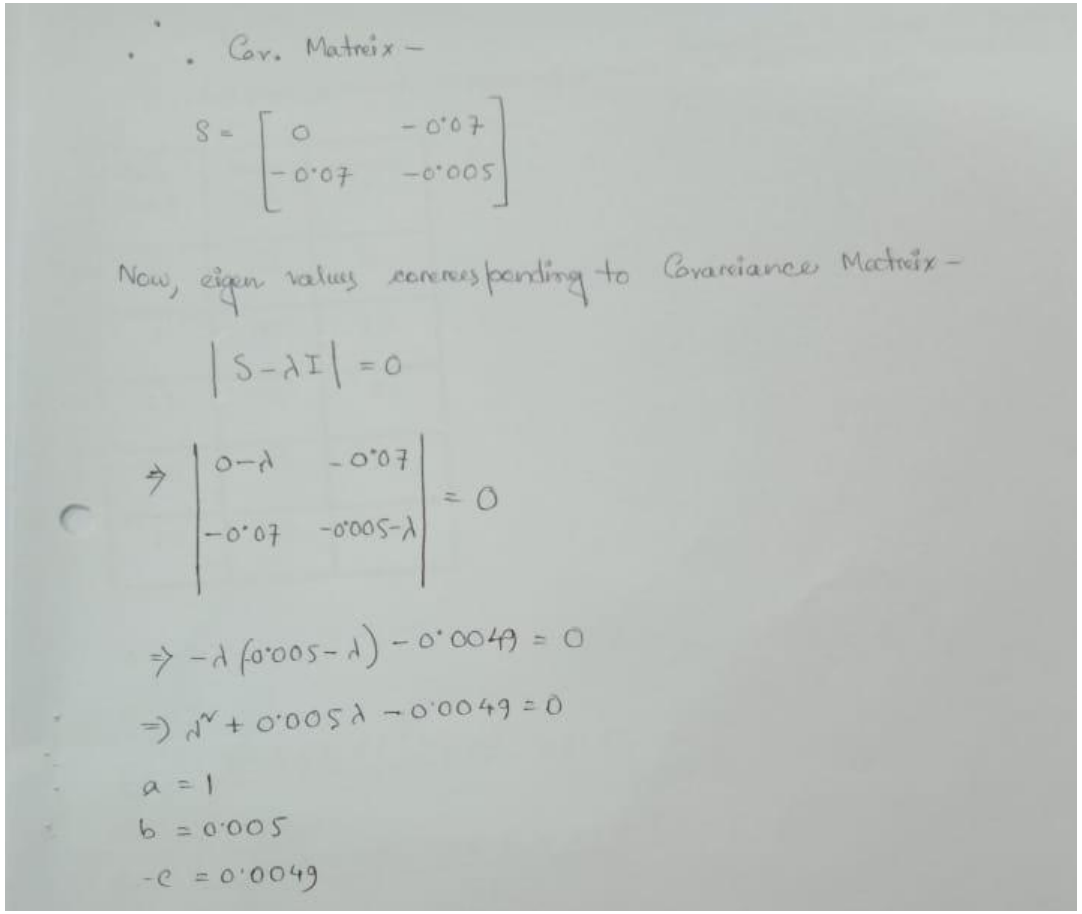
$$= \frac{-0.28}{4} = -0.07$$

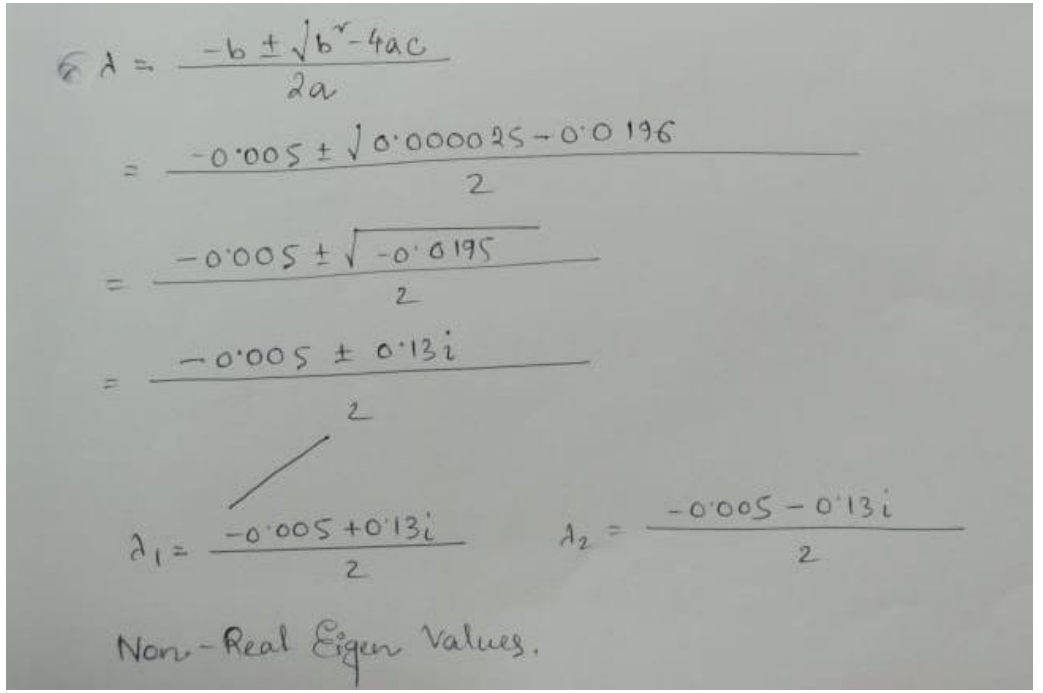
$$Cov(s,s) = \frac{1}{5-1}\{(-10.52+10.69) + (-12.45+10.69) + (-10.9+10.69) + (-9.7+10.69) + (-9.9+10.69)\}$$

$$= \frac{1}{4}\{0.17 - 1.76 - 0.21 + 0.99 + 0.79\}$$

$$= \frac{-0.02}{4} \simeq -0.005.$$

Worked Examples





Worked Examples

Q. 7 Form covariance matrix:

Variable	Ex_1	Ex_2	Ex_3
A	10	5	3
b	12	9	3

Solution:

$n = 2$

$N = 3$

$\bar{A} = \frac{10+5+3}{3} = 6$

$\bar{B} = \frac{12+9+3}{3} = 8$

$$\begin{bmatrix} (x,x) & (x,y) \\ (y,x) & (y,y) \end{bmatrix}$$

$Cov(x,x) = \frac{1}{3-1}\{(10-6)^2 + (5-6)^2 + (3-6)^2\}$

$= \frac{1}{2}\{16+1+9\} = \frac{26}{2} = 13$

$Cov(y,x) = Cov(x,y) = \frac{1}{3-1}\{(10-6)(12-8) + (5-6)(9-8) + (3-6)(3-8)\}$

$= \frac{1}{2}\{16+1+15\}$

$= \frac{32}{2} = 16$

$Cov(y,y) = \frac{1}{3-1}\{(12-8)^2 + (9-8)^2 + (3-8)^2\}$

$= \frac{1}{2}\{16+1+25\} = \frac{42}{2} = 21$

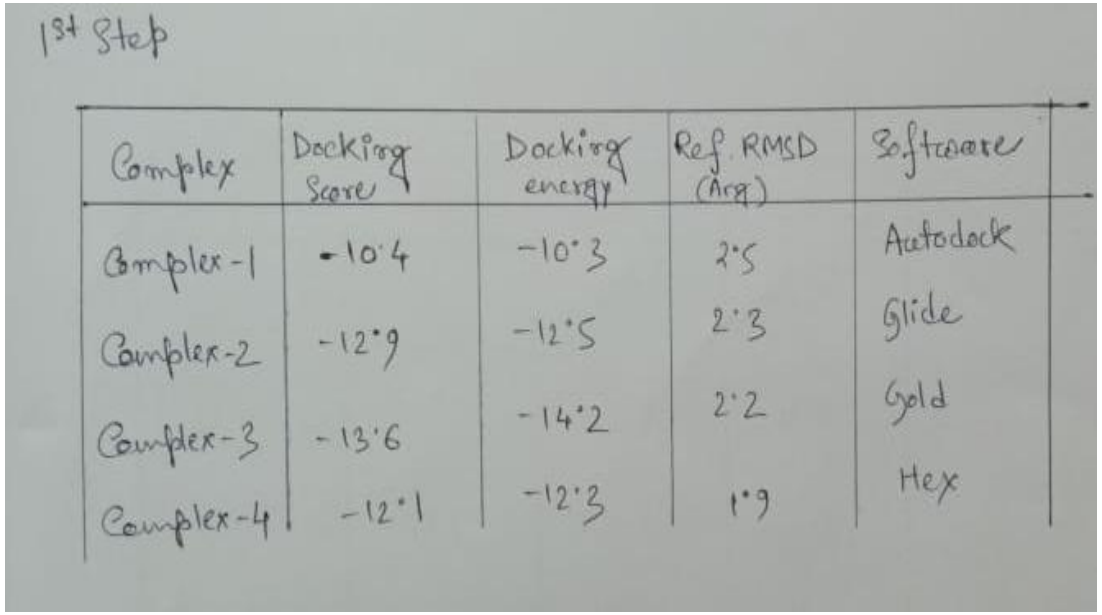
$\therefore$ Cov. matrix $= \begin{bmatrix} 13 & 16 \\ 16 & 21 \end{bmatrix}$

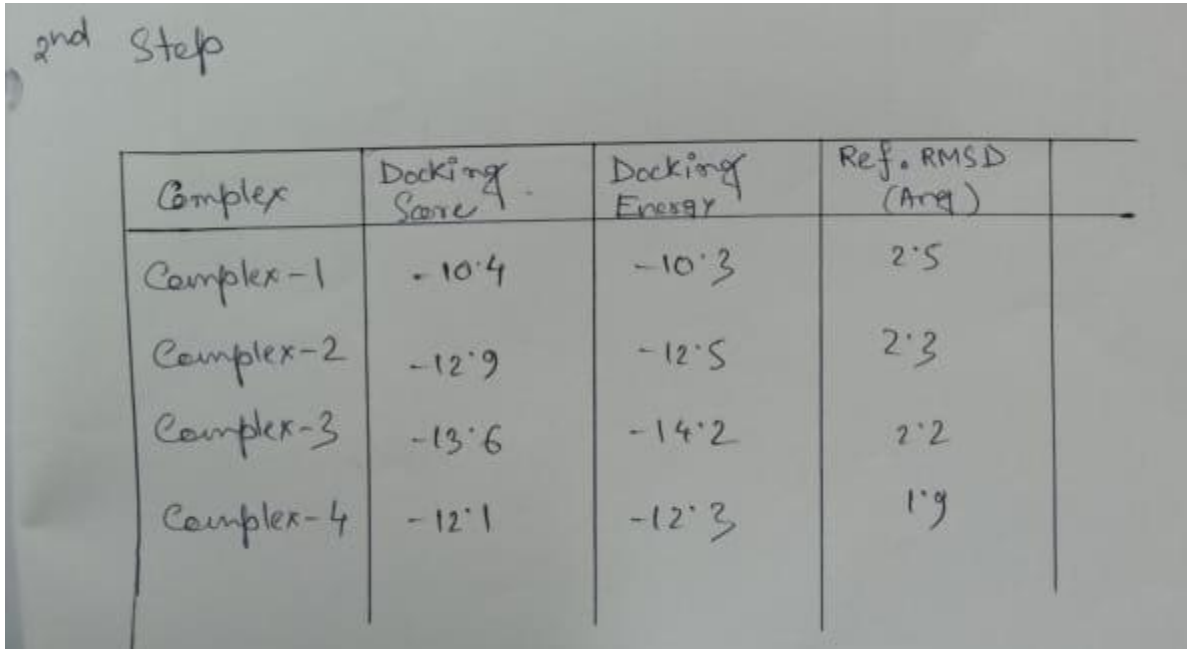
Worked Examples

Q. 8 Use feature selection for optimizing the parameters.

Solution:

Complex	**Docking Score**	**Docking Energy**	**Ref. RMSD (Ang)**	PDB pd	Software
Complex 1	-10.4	-10.3	2.5	1 BNA	Autodock
Complex 2	-12.9	-12.5	2.3	1 DNE	Glide
Complex 3	-13.6	-14.2	2.2	2 MNE	Gold
Complex 4	-12.1	-12.3	1.9	195 D	Hex





Worked Examples

Q. 9 Evaluate using PCA:

Variable	Ex_1	Ex_2	Ex_3	Ex_4	Ex_5	Ex_6
A	10	7	3	6	4	6
b	12	6	2	4	6	18

Solution:

$n = 2, \quad N = 6$

$\bar{a} = \frac{10+7+3+6+4+6}{6} = 6$

$\bar{b} = \frac{12+8+2+4+6+18}{6} = 8$

Now, Cov. Mat. –

$$S = \begin{bmatrix} Cov(a,a) & Cov(a,b) \\ Cov(b,a) & Cov(b,b) \end{bmatrix}$$

$Cov(a,a) = \frac{1}{6-1}\{(10-6)^2 + (7-6)^2 + (3-6)^2 + (6-6)^2 + (4-6)^2 + (6-6)^2\}$

$= \frac{1}{5}\{16+1+9+0+4+0\} = \frac{1}{5}\{30\} = 6$

$Cov(b,b) = \frac{1}{6-1}\{(12-8)^2 + (6-8)^2 + (2-8)^2 + (4-8)^2 + (6-8)^2 + (18-8)^2\}$

$= \frac{1}{5}\{16+4+36+16+4+100\} = \frac{1}{5}\{176\} = 35.2$

$Cov(a,b) = Cov(b,a)$

$= \frac{1}{5}\{(10-6)(12-8) + (7-6)(6-8) + (3-6)(2-8) + (6-6)(4-8) + (4-6)(6-8) + (6-6)(18-8)\}$

$= \frac{1}{5}\{-16-2+24+4\} = \frac{10}{5} = 2$

Worked Examples

$\therefore S = \begin{bmatrix} 6 & 2 \\ 2 & 35.2 \end{bmatrix}$

Now, Eigen values of cov. matrix $= |S - \lambda I| = 0$

$$\begin{vmatrix} 6-\lambda & 2 \\ 2 & 35.2-\lambda \end{vmatrix} = 0$$

$\Rightarrow (6-\lambda)(35.2-\lambda) - 4 = 0$

$\Rightarrow 211.2 - 6\lambda - 35.2\lambda + \lambda^2 - 4 = 0$

$\Rightarrow \lambda^2 - 41.2\lambda - 207.2 = 0$

approx $\lambda_1 = 45.5 \qquad \lambda_2 = -4.54$

Now, eigen vector corr to $\lambda_1 \rightarrow$

$$(S - \lambda_1 I) U_1 = 0$$

$$\begin{bmatrix} 6-\lambda_1 & +2 \\ 2 & 35.2-\lambda_1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

$(6-\lambda_1)u_1 + 2u_2 = 0$

$2u_1 + (35.2 - \lambda_1)u_2 = 0$

$\Rightarrow \frac{u_1}{-2} = \frac{u_2}{(6-\lambda_1)} = t$

@ $t_1 = 1 \Rightarrow u_1 = -2$ & $u_2 = 6 - \lambda_1 = 6 - 45.5 = 39.5$

$$U_1 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -2 \\ -39.5 \end{bmatrix}$$

Now, normalization of U_1 :—

$$e_1 = \begin{bmatrix} -2/\sqrt{(-2)^2 + (-39.5)^2} \\ -39.5/\sqrt{(-2)^2 + (-39.5)^2} \end{bmatrix} = \begin{bmatrix} -2/39.55 \\ -39.5/39.55 \end{bmatrix}$$

$$e_1 = \begin{bmatrix} -0.05 \\ -1 \end{bmatrix}$$

Worked Examples

Now, reduction of the dimension of the old data set —

	Ex_1	Ex_2	Ex_3	Ex_4	Ex_5	Ex_6
Pc_1	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}

Now, $P_{11} = e_1^T \begin{bmatrix} 10 & -6 \\ 12 & -8 \end{bmatrix}$

$= \begin{bmatrix} -0.05 & -1 \end{bmatrix} \begin{bmatrix} 4 \\ -4 \end{bmatrix} = 4.08$

$P_{12} = \begin{bmatrix} -0.05 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} = 1.95$

$P_{13} = \begin{bmatrix} -0.05 & -1 \end{bmatrix} \begin{bmatrix} -3 \\ -6 \end{bmatrix} = 6.15$

$P_{14} = \begin{bmatrix} -0.05 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ -4 \end{bmatrix} = 4$

$P_{15} = \begin{bmatrix} -0.05 & -1 \end{bmatrix} \begin{bmatrix} -2 \\ -2 \end{bmatrix} = 2.1$

$P_{16} = \begin{bmatrix} -0.05 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 10 \end{bmatrix} = -10$

Therefore, the reduced data set is —

	Ex_1	Ex_2	Ex_3	Ex_4	Ex_5	Ex_6
Pc_1	4.08	1.95	6.15	4	2.1	−10

Worked Examples

Q. 10 From the data given below related to never married, married, divorced and widowed, state whether marital status and qualifications relate to each other or not using Chi-square test.

Solution:

Table of Observed Values

M-St. / Qualif^n	Mid. School	High School	Bach.	Mast.	Ph.D.	Total
Never Married	18	36	21	9	6	90
Married	12	36	45	36	21	150
Divorced	6	9	9	3	3	30
Widowed	3	9	9	6	3	30
Total	39	90	84	54	33	300

Now, → R×C / 300

Null Hypo. – that there is no rl^n. b/w M-St. & qualif^n

Alternate Hypo. – that there is sig. rl^n b/w the mar st. & qualif^n

Let, significance level (α) = 0.05

Table for Expected Values

M-St. / Qualif^n	Mid. Sc.	High. Sc.	Bach.	Mast.	Ph.D.
Nev. Married	11.7	27	25.2	16.2	9.9
Married	19.5	45	42	27	16.5
Divorced	3.9	9	8.4	5.4	3.3
Widowed	3.9	9	8.4	5.4	3.3

Now, deviation = $\frac{(\text{observed} - \text{expected})^2}{\text{expected}}$

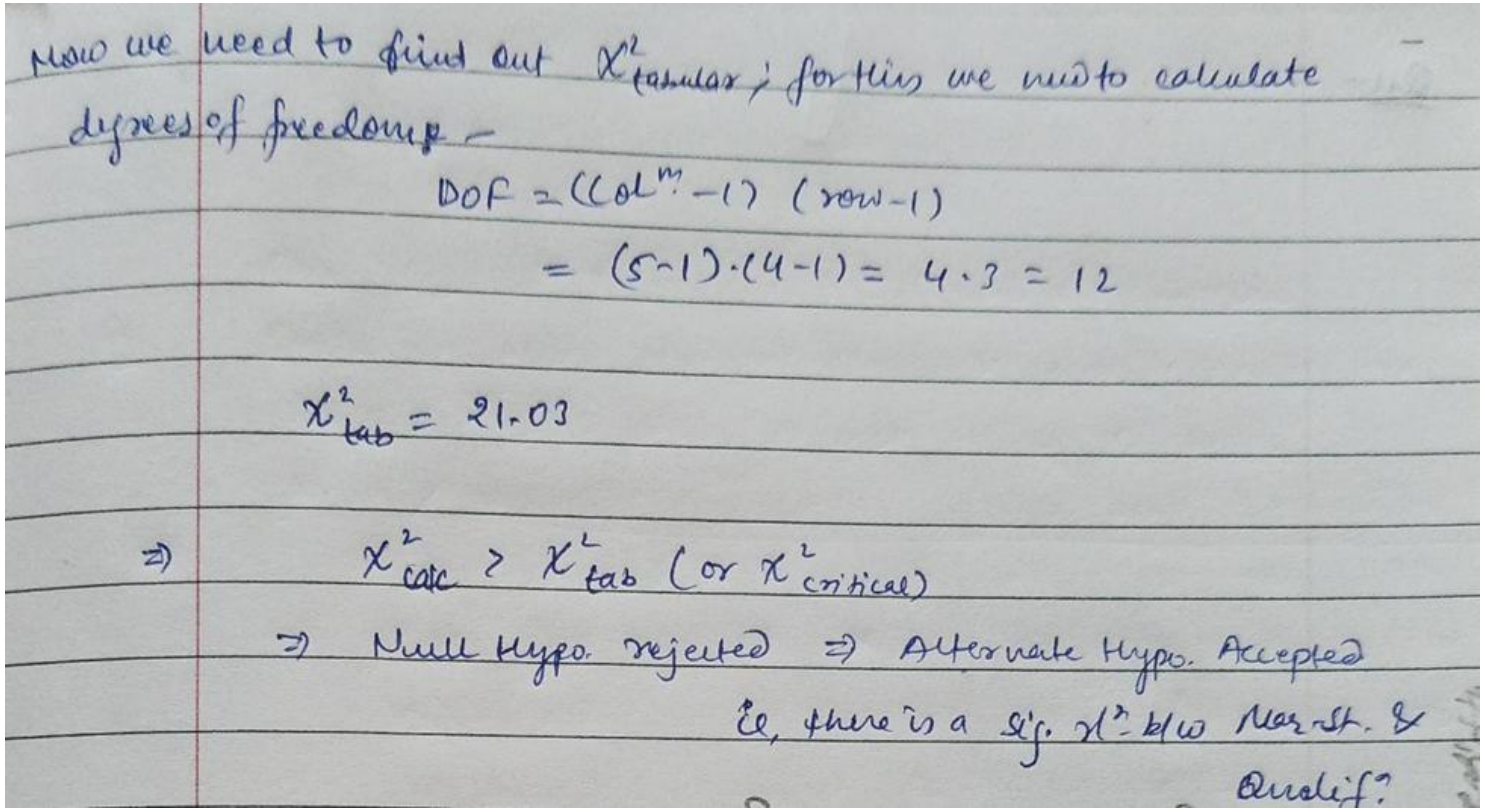
Worked Examples

Now, we prepare another table –

obs. Val. (O)	Exp. Val. (E)	(O-E)	$(O-E)^2$	$\frac{(O-E)^2}{E}$
18	11.7	6.3	39.69	3.39
36	27	9	81	3
21	25.2	-4.2	17.64	0.7
9	16.2	-7.2	51.84	3.2
6	9.9	-3.9	15.21	1.53
12	19.5	-7.5	56.25	2.88
36	45	-9	81	1.8
45	42	3	9	0.2
36	27	9	81	3
21	16.5	4.5	20.25	1.22
6	3.9	2.1	4.41	1.13
9	9	0	0	0
9	8.4	0.6	0.36	0.04
3	5.4	-2.4	5.76	1.06
3	3.3	-0.3	0.09	0.02
3	3.9	-0.9	0.81	0.20
9	9	0	0	0
9	8.4	0.6	0.36	0.04
6	5.4	0.6	0.36	0.06
3	3.3	-0.3	0.09	0.02

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

$$\chi^2_{calc.} = 23.57$$



Now we need to find out $\chi^2_{tabular}$; for this we need to calculate degrees of freedom –

$$DOF = (Col^{n} - 1)(row - 1)$$

$$= (5-1)\cdot(4-1) = 4 \cdot 3 = 12$$

$$\chi^2_{tab} = 21.03$$

$\Rightarrow$ $\chi^2_{calc} > \chi^2_{tab}$ (or $\chi^2_{critical}$)

$\Rightarrow$ Null Hypo. rejected $\Rightarrow$ Alternate Hypo. Accepted

i.e, there is a Sig. χ^2 b/w Nar-St. & Qualif?

Exercises

Question 4: Which of the following can be the first two principal components after applying PCA?

1) (0.5, 0.5, 0.5, 0.5) and (0.71, 0.71, 0, 0)
2) (0.5, 0.5, 0.5, 0.5) and (0, 0, -0.71, -0.71)
3) (0.5, 0.5, 0.5, 0.5) and (0.5, 0.5, -0.5, -0.5)
4) (0.5, 0.5, 0.5, 0.5) and (-0.5, -0.5, 0.5, 0.5)

a) 1 and 2 b) 1 and 3 c) 2 and 4 d) 3 and 4

Question 5: A computational chemistry data is summarized as follows:
Compute the Eigen values corresponding to the covariance matrix formed.

Bond length	**Bond angle**	**Remark**
2	102	Yes
3	107	Yes
2.7	156	No
1.47	165	No
1.5	148	Yes

Question 6: Use feature selection for optimizing the parameters.

Composite	**HOMO**	**LUMO**	**E**	Point Group
Ge-O	-1.2	-2.4	-1.2	C1
Si-O	-2.9	-3.2	-0.3	C2
Cr-Sr-Ca-O	-1.1	-2.1	-1.0	C1
Ni-Zn-Fe	-3.2	-3.9	-0.7	C2

Question 7: Use feature selection method to optimize the given data.

	sepal_len	sepal_wid	petal_len	petal_wid	class
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

Question 8: Find out if the following data have any relationship between them or not. (Hint. Chi-Square test)

Species	Docking Score	Docking Energy	Total
DNA-ligand	-10.5	-10.7	-21.2
Protein-Ligand	-10.2	-10.4	-20.6
Protein-DNA	-10.7	-10.9	-21.6
Protein-protein	-10.4	-10.6	-21.0

Question 9: Find out if the following data have any relationship between them or not.

Composite	Interaction energy	Band gap	Total
Si-O-Ge	-10.5	-1.7	-11.2
Ni-Zn-Fe	-10.2	-1.4	-11.6
La-Sr-O	-10.7	-1.9	-11.6
Cd-S-O	-10.4	-1.6	-12

Question 10: Given two datasets in regard to petal length and sepal length. Compute the scatter matrix between the classes.

Petal length	Petal width	Class
5.2	2.3	Rose
5.0	1.9	Dahlia
5.2	2.0	Hibiscus
5.4	2.3	Sunflower
5.1	1.8	Lily

Sepal length	Sepal width	Class
6.7	3.0	Rose
6.3	2.5	Dahlia
6.5	3.0	Hibiscus
6.2	3.4	Sunflower
5.9	3.0	Lily

References

- Mitchell, T.M. (1997). *Machine Learning*. McGraw Hill.

➢ Covers the field of machine learning, which is the study of algorithms that allow computer programs to automatically improve through experience.★ ★★ ★

- Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer.

➢ This book reflects these developments while providing a grounding in the basic concepts of pattern recognition and machine learning.★ ★★ ★

- Hastie, T., Tibshirani, R. Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

➢ Describes the important ideas in a variety of fields such as medicine, biology, finance, and marketing in a common conceptual framework. While the approach is statistical, the emphasis is on concepts rather than mathematics.★ ★★

- Shwartz, S.S., and Davis, S.B. (2014). *Understanding Machine Learning: From Theory to Algorithm*. Cambridge University Press.

➢ Introduces machine learning and provides a theoretical account of the fundamentals underlying machine learning and the mathematical derivations that transform these principles into practical algorithms. ★ ★★

Thank You