27/02/23

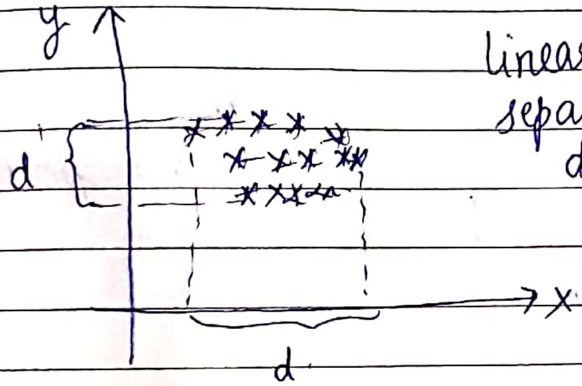
ML – Curse of Dimensionality Reduction – Performance
L Visualization

Feature selection
→ Forward selection
→ Backward Reduction

Feature Extraction
→ PCA
→ LDA
→ tsne.

Feature Engineering
- Feature Transformation
- FS
- FE
- Feature Construction

Principal Component Analysis (PCA).



linearly separable data.

$d \gg d'$

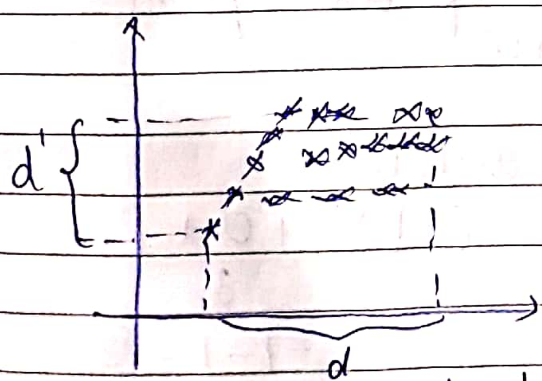
Variance acc. to y-axis is very less. y can be ignored.

Variance : Spread of data.

more the spread, higher the variance.

$$\text{Variance} = \frac{\sum (X_i - \bar{X})^2}{n}$$

Variance $\propto$ Spread.



Both the features have to be considered.

$d = d'$

PCA → shifting the axis of data points

No. of PC $\leq$ N

Our motive : Maximize variance
because we will select that feature which has higher spread.

Another factor we can use is Standard deviation $= \sqrt{\text{variance}}$

→ Projection Unit vector $= \dfrac{\vec{u} \cdot \vec{x}}{|u|}$

To calculate projection at any point. $= \vec{u} \cdot \vec{x}$

$= \vec{u} \cdot x_i$ → used in PCA

Projection of Mean Point, $x_m = \vec{u} \cdot x_m$.

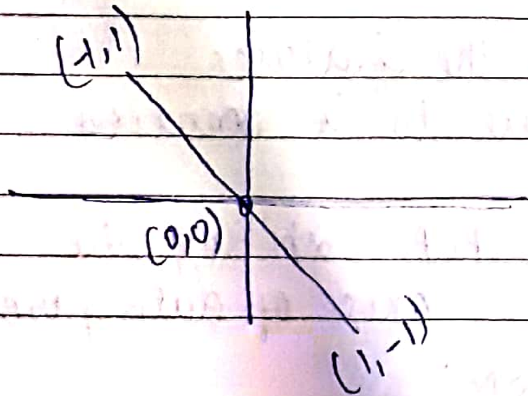
$$\vec{u} \cdot x_i = u^T \cdot x_i.$$

we calculate : $[u^T \cdot x_1], [u^T \cdot x_2], \text{-----} [u^T \cdot x_n]$ for all points.

$$\boxed{\text{Variance} = \frac{\sum_{i=1}^{n} (u^T \cdot x_i - u^T \cdot x_m)^2}{n}}$$

Our motive is to increase Variance.

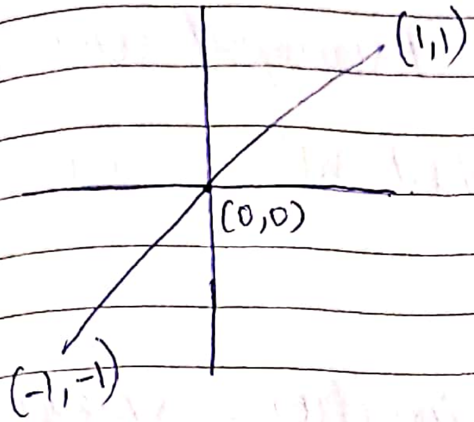
Variance doesn't tell direction of data points.
∴ We need Covariance!



Covariance

$$= \frac{(-1 \times 1) + 0 + (1 \times -1)}{3}$$

$$= \frac{-1-1}{3} = \frac{-2}{3}$$



$$\text{Covariance} = \frac{(1\times1)+0+(-1\times-1)}{3}$$

$$= \frac{1+1}{3} = \frac{2}{3}$$

Co-variance tells the alignment of data points

Covariance Matrix

$Cov(x_2, x_1)$
$= Cov(x_1, x_2)$

$$\begin{array}{cccc} & x_1 & x_2 & x_3 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \multicolumn{3}{c}{\begin{bmatrix} Cov(x_1,x_1) & Cov(x_1,x_2) & Cov(x_1,x_3) \\ Cov(x_2,x_1) & Cov(x_2,x_2) & Cov(x_2,x_3) \\ Cov(x_3,x_1) & Cov(x_3,x_2) & Cov(x_3,x_3) \end{bmatrix}} \end{array}$$

$Cov(x_1, x_1) = Var(x_1)$
$Cov(x_2, x_2) = Var(x_2)$

$$\begin{array}{ccc} & x_1 & x_2 \\ \begin{matrix} x_1 \\ x_2 \end{matrix} & \multicolumn{2}{c}{\begin{bmatrix} Var(x_1) & Cov(x_1,x_2) \\ Cov(x_2,x_1) & Var(x_2) \end{bmatrix}} \end{array}$$

1/3/2023

Matrix Transformation

Coordinate System : Collection of infinite coordinates.
Eigen Value : value corresponding to Eigen vector
Eigen Vector : a point whose direction doesn't change on applying any Transformation. Only magnitude may vary. At this point Variance is maximum.

The Eigen values are the Principal Components

→ The Eigen value with highest value is called PC_1

→ PC_2 and so on...

→ Covariance matrix indicates direction & spread of data.

→ Largest eigen vector always points to the largest spread of the data and its magnitude represents the Eigen Value

Eg:-

X_1	X_2	target
4	11	
8	4	
13	5	
7	14	

① Calculate Mean

$$\bar{X_1} = \frac{4+8+13+7}{4} = 8$$

$$\bar{X_2} = \frac{11+4+5+14}{4} = 8.5$$

② Find Covariance Matrix

$$\begin{array}{c c} & \begin{array}{cc} X_1 & X_2 \end{array} \\ \begin{array}{c} X_1 \\ X_2 \end{array} & \begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Cov(X_2, X_2) \end{bmatrix} \end{array}$$

$$Cov(X_1, X_1) = \frac{\sum_{i=1}^{n} (X_i - \bar{X}_1)^2}{n-1}$$

$$\frac{\sum (X_1 - \bar{x})(X_1 - \bar{x})}{4-1}$$

$$\frac{(8-8)^2 + (13-8)^2 + (7-8)^2 + (4-8)(4-8)}{3} = \frac{\cancel{4 \times 4}}{3} = \frac{16}{3} = 5.3\ 14$$

$$Cov(X_1, X_2) = \frac{\sum (X_1 - \bar{x}_1)(X_2 - \bar{X}_2)}{4-1}$$

$$= \frac{(4-8)(11-8.5) + (8-8)(4-8.5) + (13-8)(5-8.5) + (7-8)(14-8.5)}{3}$$

$$= \frac{(-4)(3) + 0 + 5 \times (-3.5) + (-1)(5.5)}{3}$$

$$= \frac{-10 - 17.5 - 5.5}{3} = -11$$

Covariance Matrix:

$$Covar(X_1, X_2) \begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix}$$

Finding Eigen vectors:

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

$$\begin{bmatrix} 14 & -11 \\ -11 & 23 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 0$$

$$\left| \begin{bmatrix} 14-\lambda & -11 \\ -11 & 23-\lambda \end{bmatrix} \right| = 0$$

$$(14-\lambda)(23-\lambda)-11^2$$

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

$\lambda_1 = 30.38 \rightarrow PC_1$ will have greater impact on data.

$\lambda_2 = 6.61 \rightarrow PC_2$

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

$$e_1 = \begin{bmatrix} 0.5574 \\ -0.8303 \end{bmatrix}$$

$$\begin{bmatrix} 14-\lambda_2 & -11 \\ -11 & 23-\lambda_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = 0$$

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

$\rightarrow$ Normalize the eigen vectors

Take Transpose of e_1:

$$\begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{11} - \overline{X_1} \\ X_{21} - \overline{X_2} \end{bmatrix}$$

$$= \begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} 4-8 \\ 11-8.5 \end{bmatrix}$$

$$\begin{bmatrix} 0.5574 & -0.8303 \end{bmatrix} \begin{bmatrix} -4 \\ 2.5 \end{bmatrix} = -4.30535$$

$\therefore$ new feature corresponding to (X_{11}, X_{21})

$= -4.30535$

$$\begin{bmatrix} 0.5579 & -0.8303 \end{bmatrix} \begin{bmatrix} X_{12} - \bar{X}_1 \\ X_{22} - \bar{X}_2 \end{bmatrix}$$

$$\begin{bmatrix} 0.5579 & -0.8303 \end{bmatrix} \begin{bmatrix} 8-8 \\ 4-8.5 \end{bmatrix}$$

new feature = 3.7361
corresponding to (X_{21}, X_{22})

$\xrightarrow{}$ 1D

2D X_1	X_2	new feature
4	11	-4.30535
8	4	3.7361
13	5	5.6922
7	14	-5.1238

Linear Discriminant Analysis (LDA) 13/03/23

→ used for supervised learning. used to separate 2 classes
→ The variance between classes should be max.
→ Reduce the dimensionality.

→ LDA does not change location but provides more class separability

→ PCA is used for unsupervised learning.
→ LDA is used for max. separability b/w 2 classes by creating a new axis and projecting the data on that axis

Fisher Discriminant Ratio

used for finding class separability

$$\boxed{J(m) = \frac{|\mu_1 - \mu_2|^2}{\tilde{S}_1^2 + \tilde{S}^2}} \begin{matrix} \rightarrow max \\ \rightarrow min \end{matrix}$$

$S_W \rightarrow$ Scatter matrix within class.
$S_B \rightarrow$ class scatter matrix

Our aim is to maximise the Fisher Discrimant Ratio to ~~a~~ find the axis for LDA

$$S_W = \sum_{i=1}^{C} S_i$$

C = No. of classes

NLP → Sentiment Analysis

Algorithm:

→ Tokenization
→ Feature Extraction
→ Classification.

Boolean Multinomial NB
(Binarized)

Benoulli NB : if word is present, probability = p
" " " absent, " $= ~~p-1~~$ $1-p$

Markov Chain / Markov Models /
Hidden Markov Models : sequence model whose task is to capture the sequences of words.
: They are employed on time-series data.

15/03/23

LDA

step-1 : compute the d-dimensional mean vector

step-2 : Compute the scatter matrix for each class

step-3 : Compute S_B

$$S_W = \sum_{i=1}^{L} S_i = S_1 + S_2$$

step 4 : Find the best LDA projection vector

PCA cannot be applied in non linear data

t - SNE - [t-distributed stochastic neighbours Embedding]

$$\frac{\text{Score}}{\text{Sum of all Score}}$$

To check if x_j is a neighbor of x_i

$$P_{j|i} = \frac{\exp(1 - \|x_i - x_j\|^2)\, 2\sigma_i^2}{\sum_{i \leq p} \exp(1 - \|x_i - x_j\|^2)\, 2\sigma_i^2}$$

$$q_{ij} = \frac{(i + \|y_i - y_j\|^2)^{-1}}{\sum_K \sum_{l \neq K} (1 + \|y_K - y_l\|^2)^{-1}}$$

Loss function : KL divergence

Kullback - Leibler

$$\boxed{KL(P \| Q) = \sum_{i \neq j} P_{j|i} \log \frac{P_{j|i}}{q_{j|i}}}$$

Loss function should be minimum.

Logistic Regression

: we have to find the line that classify data into 2.

: Perception

$$y = mx + b$$

$$Ax + By + C = 0$$

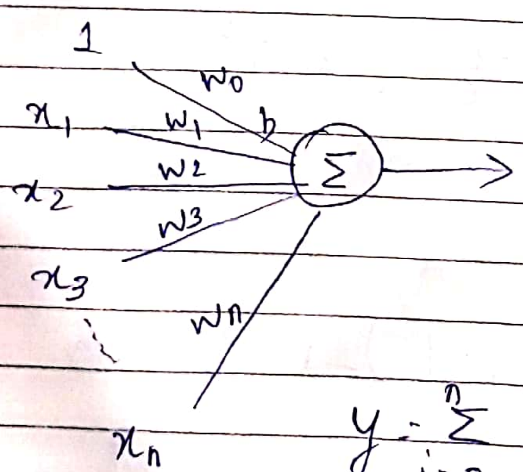
$$\frac{A}{B}x + y + \frac{C}{B} = 0$$

$$\frac{A}{B}\cancel{x + y}$$

$$Ax_1 + Bx_2 + Cx_3 + Dx_4 + Ex_5 + f = 0$$

$$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + w_5x_5 + w_0 = 0$$

$w_1, w_2, w_3, w_4 \rightarrow$ give the strength of x_1, x_2, x_3, x_4



$$W_{new} = w_0 - \eta \times \text{coordinates}$$

$$y = \sum_{i=0}^{n} w_i x_i$$

for i in range (epoch):
randomly selected points

if $x_i \in N$ and $\sum_{i=0}^{n} w_i x_i \geq 0$.

$$W_{new} = W_{old} - \eta \times x_i$$

if $x_i \in P$ and $\sum_{i=0}^{n} w_i x_i < 0$

$$W_{new} = W_{old} + \eta x_i$$

Log-Loss Function / Cross Entropy

desmos.com

$$L = \frac{-1}{n} \sum_{i=1}^{n} \left[y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i) \right]$$

Logistic Regression
LDA
PCA
t-SNE

Minimize the cost function:

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \vdots \\ \hat{y}_m \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \cdots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \cdots & X_{2n} \\ \vdots & & & & \\ x_{m1} & X_{m2} & \cdots & \cdots & X_{mn} \end{bmatrix}$$

$$= \sigma\left(\begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1m} \\ \vdots & & & \vdots \\ X_{m1} & \cdots & \cdots & X_{mn} \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix} \right)$$

Derivation to find min. value of loss function

$$\boxed{W_{new} = W_{old} - \eta \frac{(Y - \hat{Y}) X}{m}}$$

NLP

Hidden Markov Model

Forward Algorithm for Likelihood Calculation

$$HMM, \lambda = (A, B)$$
$$O = \{o_1, o_2, o_3\}$$

$$P(O \mid \lambda)$$
$$P(o_1, o_2, o_3 \mid \lambda)$$

Three steps:

1. Initialization

$$\alpha_1(j) = a_{0j} \cdot b_j(o_1) \quad \text{for } j = 1 \text{ to } N$$

2. Recursion $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) \cdot a_{ij} b_j(o_t)$

3. Termination

$$P(O \mid \lambda) = \alpha_T(q_F) = \sum_{i=1}^{N} \alpha_T(i) \cdot a_{iF}$$

a_{ij} = transition prob.

$b_j(o_t)$ = observation probability of observation o_t given state j.