PCA Algorithm

Step 1: Calculate $\underbrace{\Sigma_x}_{D\times D} \rightarrow X^T X / N$ after centering.

Step 2: Find first D' eigenvectors of Σ_x.

eigenvectors that correspond to D' largest eigenvalues

$$W = \underbrace{\begin{bmatrix} | & | & & | \\ w_1 & w_2 & \cdots & w_{D'} \\ | & | & & | \end{bmatrix}}_{D\times D'}$$

Projection Step:

$$\underbrace{z_i}_{D'\times 1} = \underbrace{W^T}_{D'\times D} \underbrace{(x_i - \overbrace{\hat{\mu}}^{\text{centering}})}_{D\times 1} \quad \forall i$$

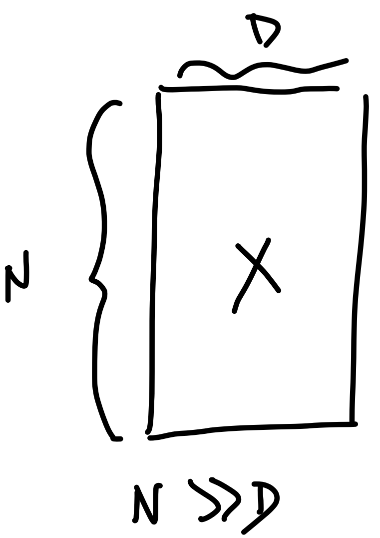
$$\hat{\mu} = \frac{\sum_{i=1}^{N} x_i}{N}$$

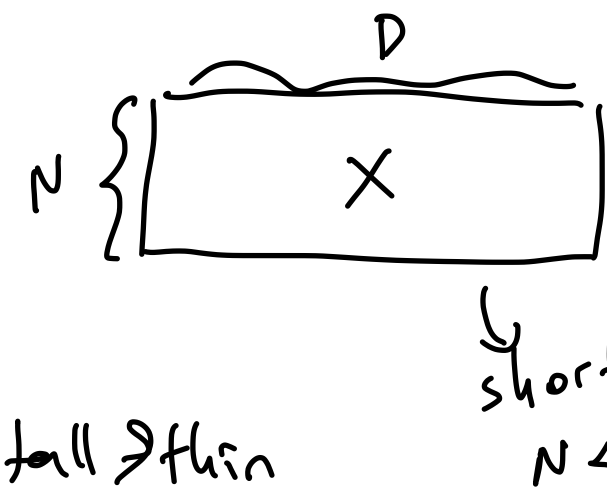
How to pick D'? Proportion of Variance Explained (PoVE)

$$\frac{X^T . X}{N} \left\{ \Sigma_x = \frac{\sum_{i=1}^{N} \underbrace{(x_i - \hat{\mu})}_{D\times 1} \underbrace{(x_i - \hat{\mu})^T}_{1\times D}}{N} \right.$$

if $N \geq D$, $\Rightarrow$ full rank

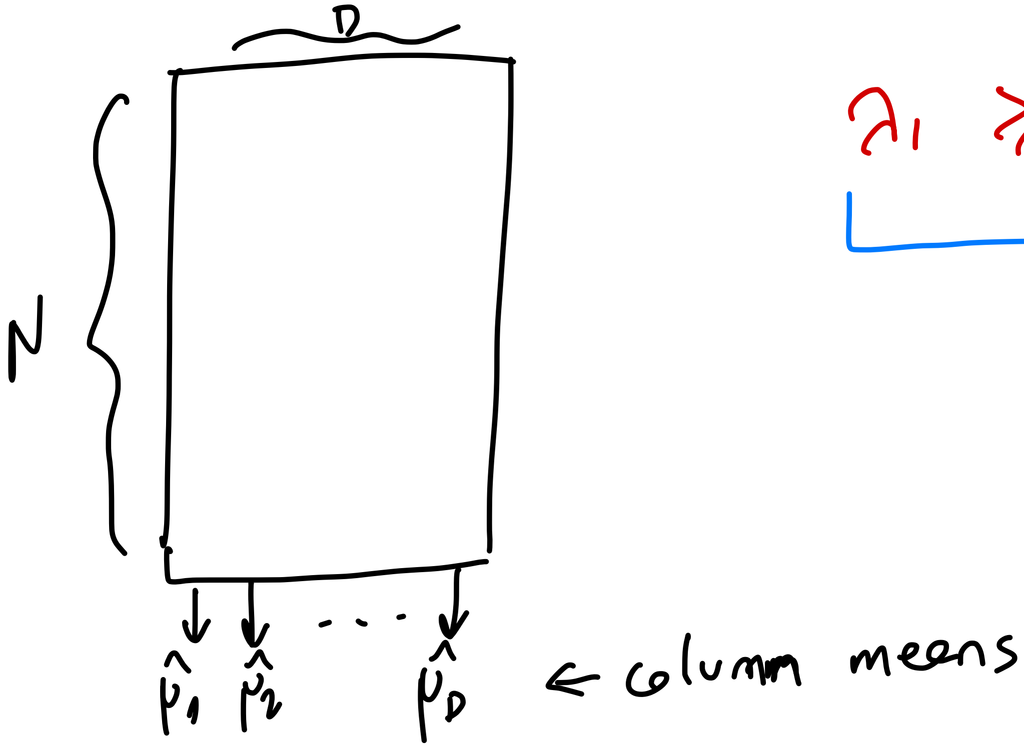
if $N < D$, $\Rightarrow$ rank-deficient





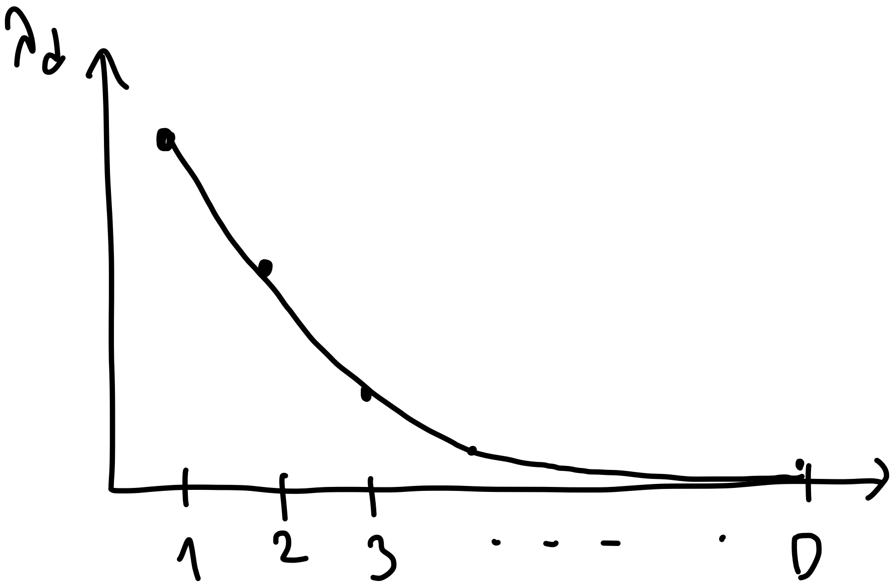
$XX^T \Rightarrow N\times N$

$X^T . X \Rightarrow D\times D$



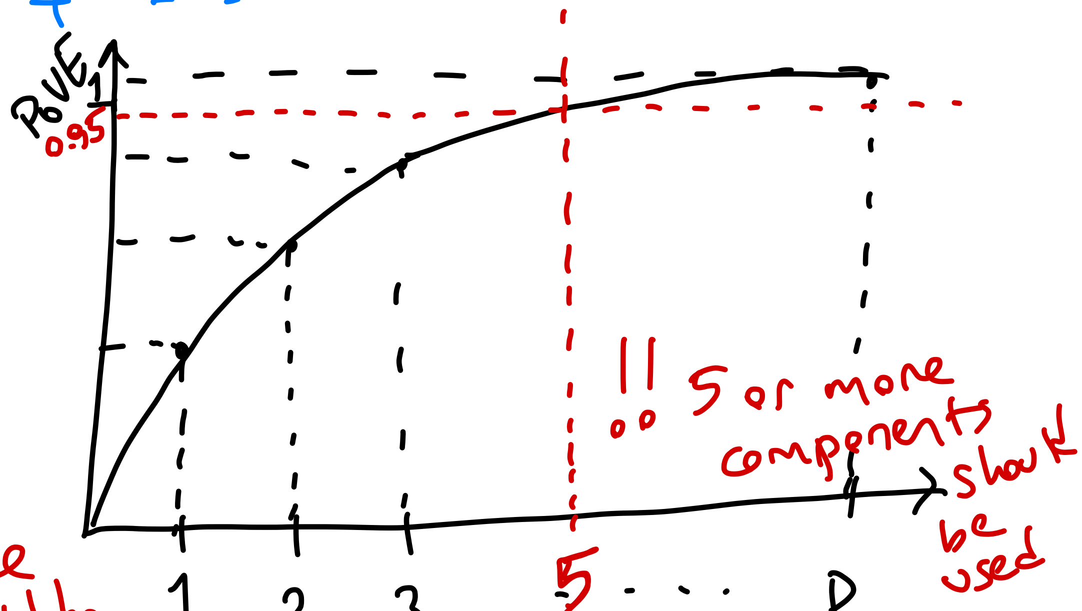
$$\underbrace{\lambda_1 \geq \lambda_2 \geq - - - -}_{\underline{\underline{D'}}} \geq \lambda_D \geq 0$$

$$\mathrm{PoVE}(D') = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_{D'}}{\lambda_1 + \lambda_2 + \cdots + \lambda_D}$$



PoVE
$\Longrightarrow$

95% of variance should be preserved.



If we use all eigen vectors:

$$W = \begin{bmatrix} | & | & & | \\ w_1 & w_2 & \cdots & w_D \\ | & | & & | \end{bmatrix}$$

$$WW^T = \underbrace{\begin{bmatrix} | & | & & | \\ w_1 & w_2 & \cdots & w_D \\ | & | & & | \end{bmatrix}}_{D \times D} \underbrace{\begin{bmatrix} - & w_1 & - \\ - & w_2 & - \\ & \vdots & \\ - & w_D & - \end{bmatrix}}_{D \times D}$$

eigenvecters are unit-norm.

$$WW^T = \underbrace{\begin{bmatrix} \overset{1}{w_1^T w_1} & w_1^T w_2 & \cdots & \overset{0}{w_1^T w_D} \\ w_2^T w_1 & \overset{1}{w_2^T w_2} & \cdots & w_2^T w_D \\ \vdots & \vdots & \ddots & \vdots \\ w_D^T w_1 & w_D^T w_2 & \cdots & \overset{1}{w_D^T w_D} \end{bmatrix}}_{D \times D} = I$$

eigenvecters are orthogonal.

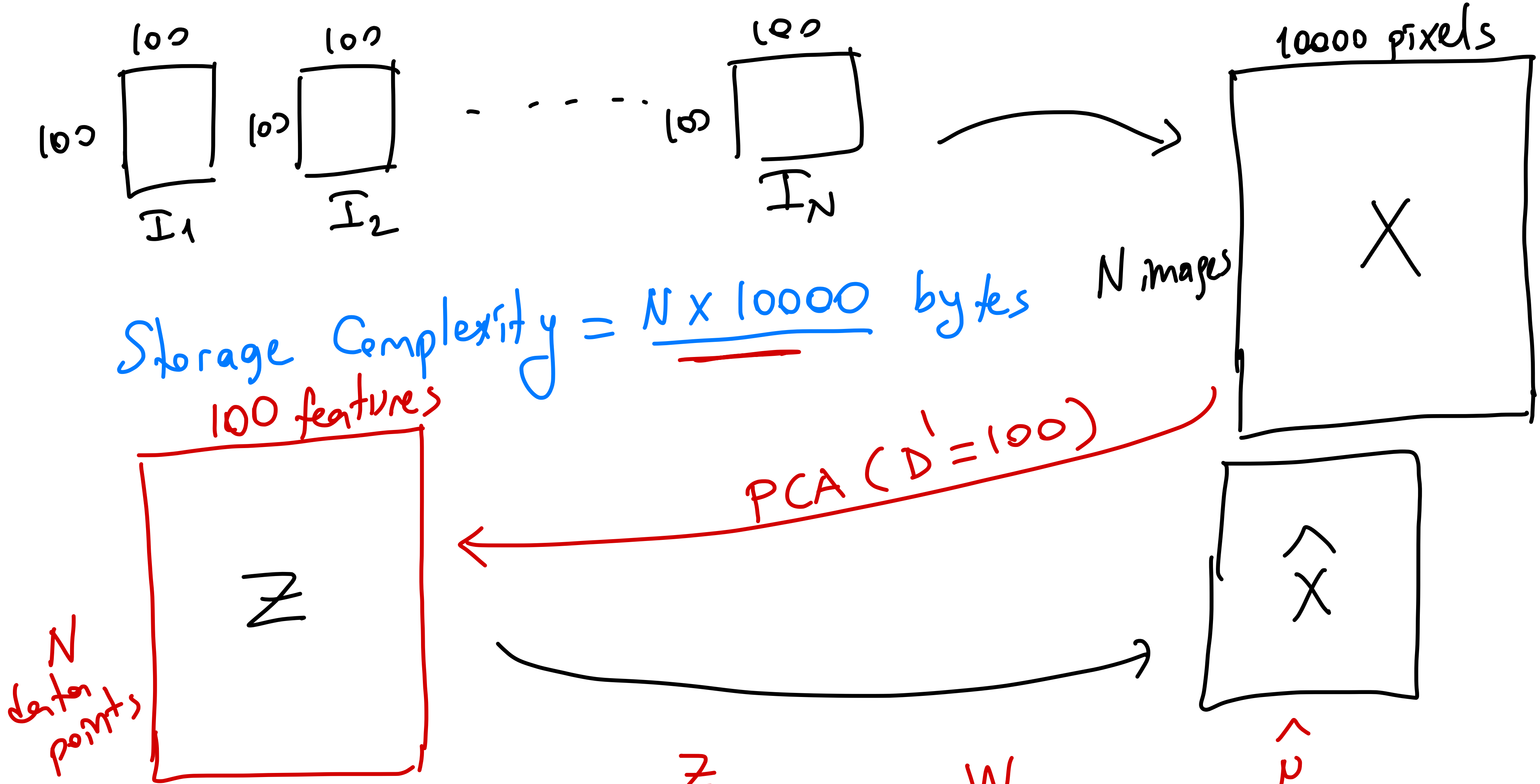
$$W z_i = \underbrace{W W^T}_{I} (x_i - \hat{\mu})$$

$$\boxed{\hat{x}_i = W z_i + \hat{\mu}}$$

$$\underset{\mathbb{R}^D}{x_i} \longrightarrow \underset{\mathbb{R}^{D'}}{z_i} \longrightarrow \underset{\mathbb{R}^D}{\hat{x}_i}$$

there will be some error if $D' < D$

$$\text{Reconstruction Error} = \sum_{i=1}^{N} \| x_i - \hat{x}_i \|_2^2$$



Storage Complexity $= \underline{N \times 10000}$ bytes

$$\text{Storage Complexity} = \overbrace{N \times 100}^{Z} + \overbrace{10000 \times 100}^{W} + \overbrace{10000 \times 1}^{\hat{\mu}}$$

$$= \left[\underline{N \times 100} + \underline{10000 \times 101}\right]$$

Principal Component Analysis vs Factor Analysis

$X \longrightarrow Z$

$Z \longrightarrow X$

$$x_d - \mu_d = \overbrace{V_{d_1}}^{\text{loadings}} z_1 + V_{d_2} z_2 + \dots + V_{dD'} z_{D'} + \epsilon_d$$

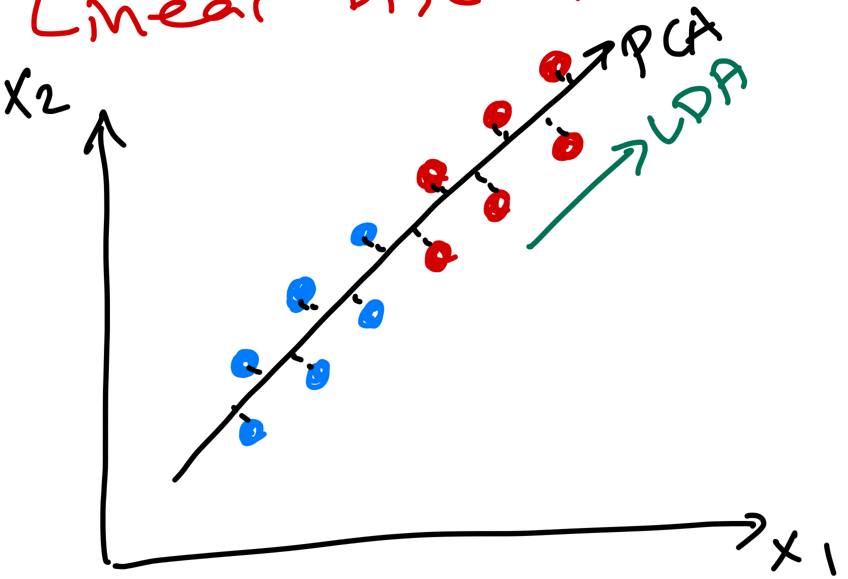
$$\underbrace{x}_{D\times 1} - \underbrace{\mu}_{D\times 1} = \underbrace{V}_{D\times D'} \cdot \underbrace{z}_{D'\times 1} + \underbrace{\epsilon}_{D\times 1}$$

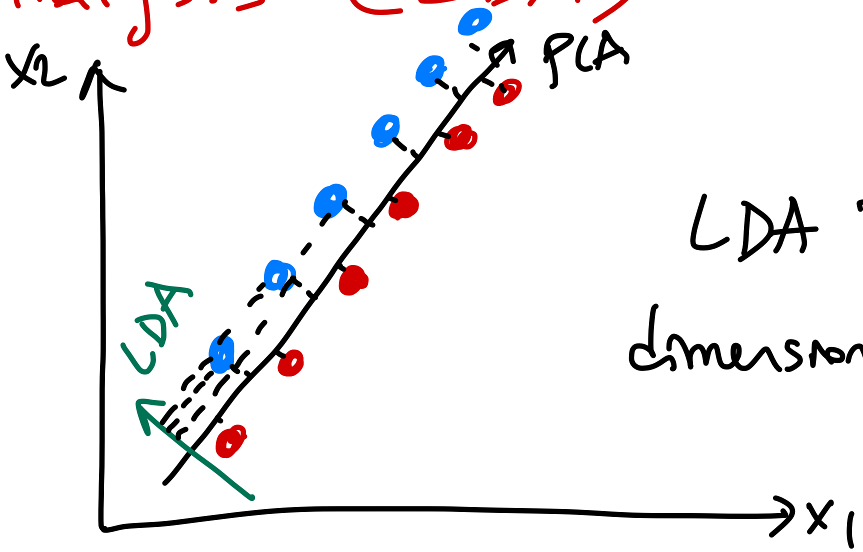
V: factor loadings.

z: hidden or latent factors

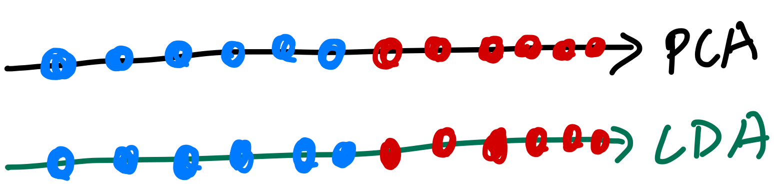
ϵ: noise term

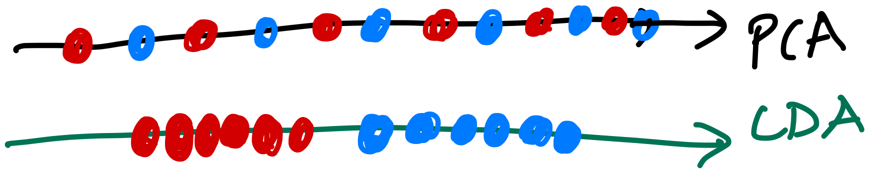
Linear Discriminant Analysis (LDA) Fisher's Discriminant Analysis (FDA)





LDA is a supervised dimensionality reduction algorithm.





$$\mathcal{X} = \{(x_i, y_i)\}_{i=1}^{N}$$

$$\begin{cases} y_i = 0 & \text{if negative} \\ y_i = 1 & \text{if positive} \end{cases}$$

$$\overbrace{z_i}^{1\times 1} = \overbrace{\underline{w}^T}^{1\times D} \cdot \overbrace{x_i}^{D\times 1}$$

$$|\hat{\mu}_1 - \hat{\mu}_2| \Rightarrow \text{as large as possible}$$

sample mean in the projected space.

$$s_1^2 + s_2^2 \Rightarrow \text{as small as possible}$$

sample variance in the projected space

$$\underbrace{\hat{\mu}_1}_{\text{semple mean of positive class}} = \frac{\sum_{i=1}^{N} z_i . y_i^{\;1}}{\sum_{i=1}^{N} y_i} = \frac{\sum_{i=1}^{N} (w^T . x_i) . y_i}{\sum_{i=1}^{N} y_i} = w^T . \underbrace{\left[\frac{\sum_{i=1}^{N} x_i . y_i}{\sum_{i=1}^{N} y_i} \right]} = w^T . \mu_1$$

sample mean of the positive class in the original space.

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{N} z_i . (1 - y_i)}{\sum_{i=1}^{N} (1 - y_i)} = \frac{\sum_{i=1}^{N} w^T . x_i (1 - y_i)}{\sum_{i=1}^{N} (1 - y_i)} = w^T . \left[\frac{\sum_{i=1}^{N} x_i (1 - y_i)}{\sum_{i=1}^{N} (1 - y_i)} \right] = w^T . \mu_2$$

$$[1 \quad 2 \quad 3 \quad 4] \Rightarrow 2.5 = \mu$$

$$[1 \quad 4 \quad 9 \quad 16] \Rightarrow 7.5 \neq (2.5)^2$$

$$3 * [1 \quad 2 \quad 3 \quad 4] - 5 \Rightarrow \hat{\mu} = 2.5 * 3 - 5 = 2.5$$

$$[3 \quad 6 \quad 9 \quad 12] - 5 \Rightarrow [\underbrace{-2 \quad 1 \quad 4 \quad 7}_{2.5}]$$

$$\underbrace{s_1^2}_{\text{scalar}} = \sum_{i=1}^{N} (z_i - \hat{\mu}_1)^2 y_i = \sum_{i=1}^{N} (w^T . x_i - \underbrace{\hat{\mu}_1}_{w^T . \mu_1})^2 . y_i$$

$$\underbrace{s_2^2}_{\text{scalar}} = \sum_{i=1}^{N} (z_i - \hat{\mu}_2)^2 . (1-y_i) = \sum_{i=1}^{N} (w^T . x_i - \underbrace{\hat{\mu}_2}_{w^T . \mu_2})^2 . (1-y_i)$$

$$s_1^2 = \sum_{i=1}^{N} (w^T . x_i - w^T . \mu_1)^2 . y_i$$

$$= \sum_{i=1}^{N} w^T . (x_i - \mu_1)(x_i - \mu_1)^T . w . y_i$$

$$= w^T . \Bigg[\underbrace{\sum_{i=1}^{N} (x_i - \mu_1)(x_i - \mu_1)^T . y_i}_{S_1}\Bigg] . w$$

$$s_2^2 = w^T . \Bigg[\sum_{i=1}^{N} \underbrace{(x_i - \mu_2)(x_i - \mu_2)^T . (1-y_i)}_{S_2}\Bigg] . w$$

$$J(w) = \frac{(\hat{\mu}_1 - \hat{\mu}_2)^2}{\underbrace{s_1^2}_{} + \underbrace{s_1^2}_{}} \Rightarrow (w^T.\mu_1 - w^T.\mu_2)^2$$

$$w^T.S_1.w + w^T.S_2.w$$

$$w^T.\underbrace{(\mu_1-\mu_2)(\mu_1-\mu_2)^T}_{S_B}.w$$

S_B = between class scatter matrix

$$J(w) = \frac{w^T.S_B.w}{w^T.\underbrace{(S_1+S_2)}_{S_W}.w}$$

S_W = within-class scatter matrix

$w^* \Rightarrow$ how to optimize w?