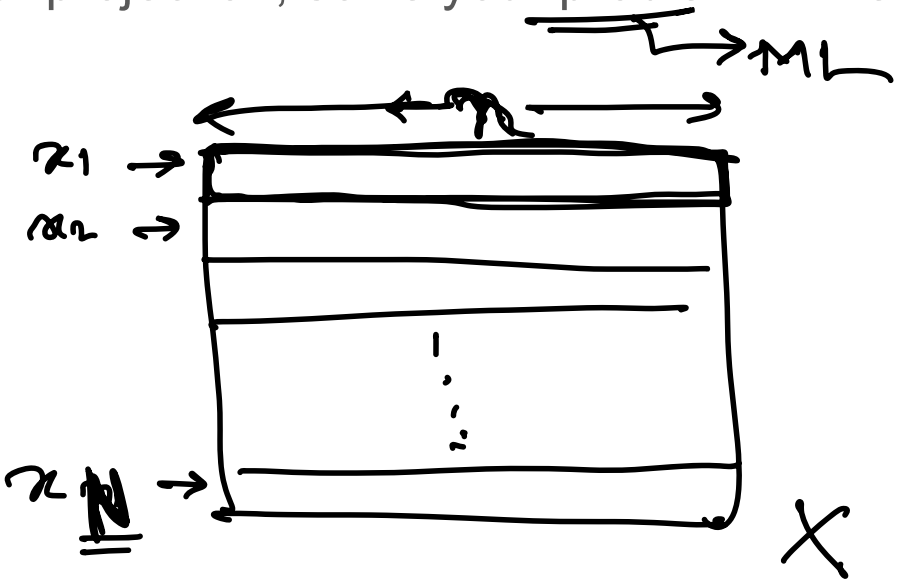
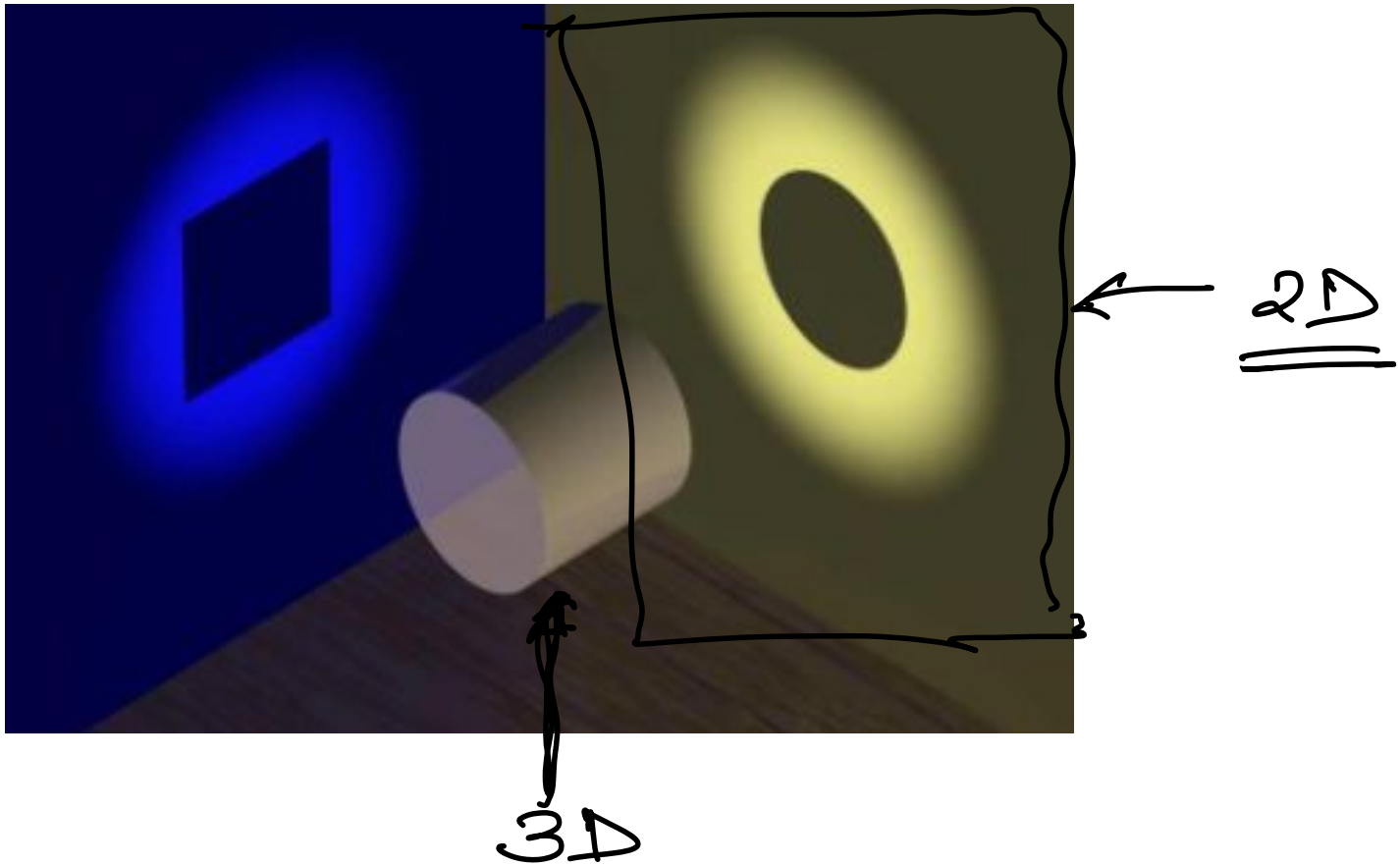
Dimensionality Reduction

- The main idea – Reduce the dimensionality of the space
- Project the n dimensional points to a k dimensional space so that
  - k << n
  - Some important properties of the data are preserved
- After projection, solve your problem in the low dimensional space

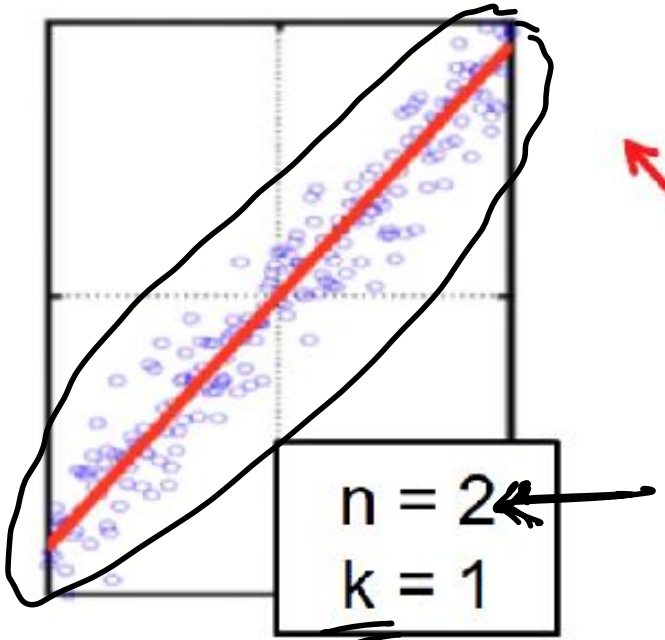


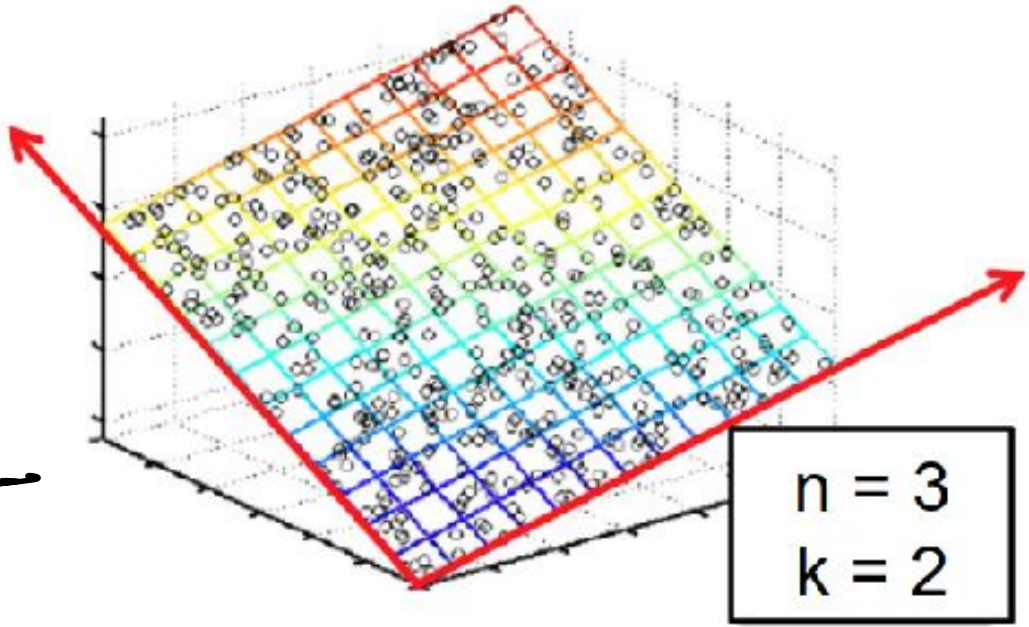

2D
3D

Dimensionality Reduction (benefits)

- Easier learning – fewer parameters
  - Less computation time for training/inference
  - Less space needed to store data
- Visualization – show high dimensional data in 2-d
- Discover “intrinsic dimensionality” of data
  - noise removal
  - high dimensional data that is actually lower dimensional

High dimensional data that is actually lower dimensional

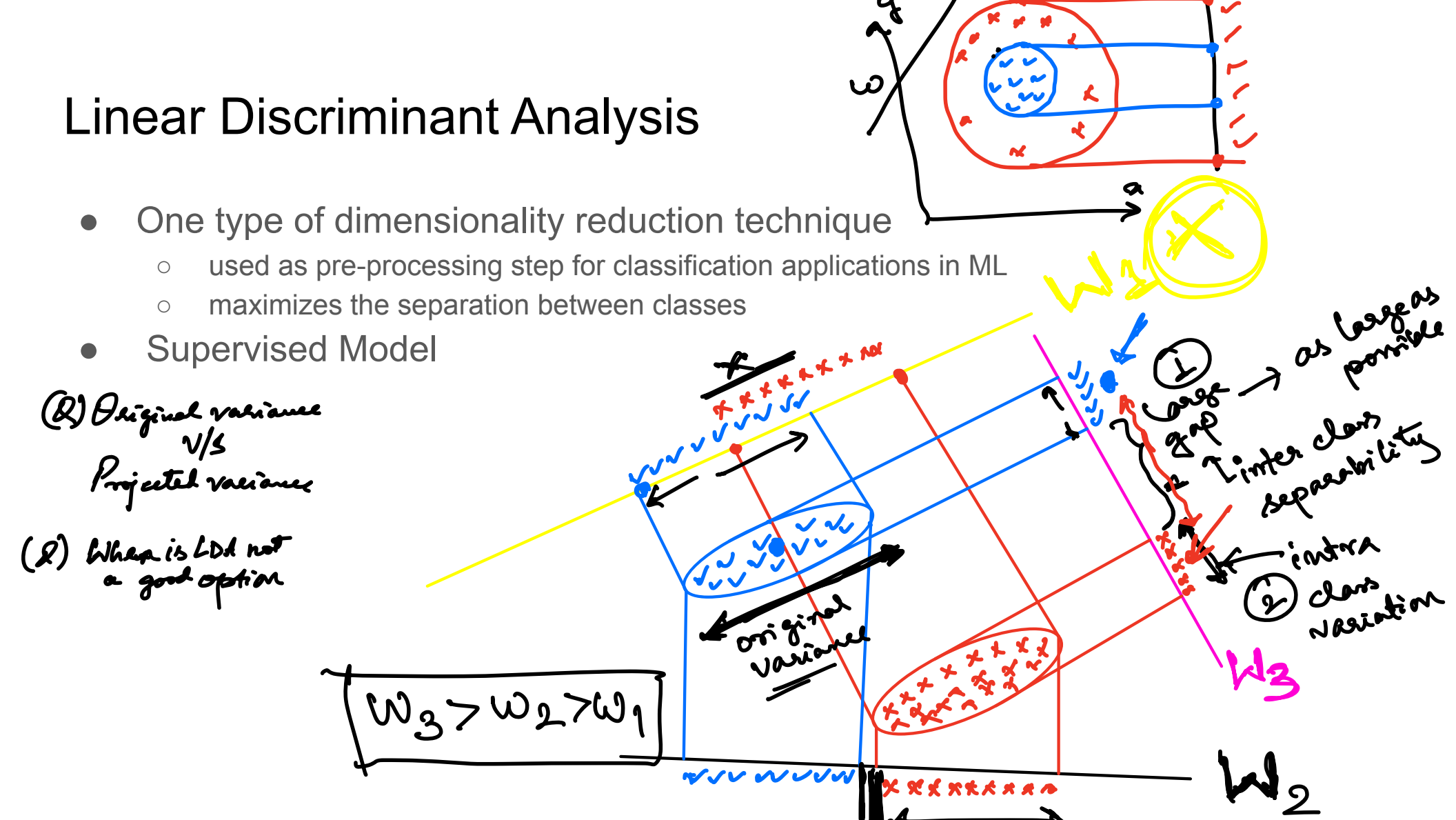




$k << n$

Linear Discriminant Analysis

- One type of dimensionality reduction technique
  - used as pre-processing step for classification applications in ML
  - maximizes the separation between classes
- Supervised Model



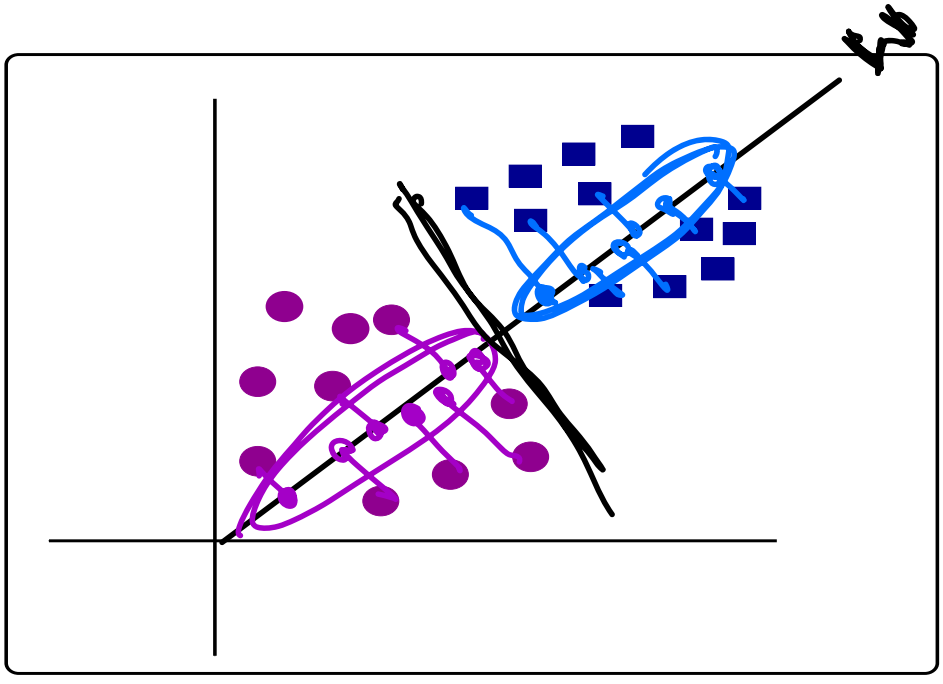
Fisher Linear Discriminant

- A linear discriminant function based classifier is:

  Decide $X \in$ C-1 if $W^T X + w_0 > 0$

- Hence One can think of the best W as the direction along which the two classes are well separated.
- We project the data along the direction W. Separation between points of different classes in the projected data is a good way to rate how good is W.
- Such a method is called Fisher Linear Discriminant.

- Consider the following 2-class example

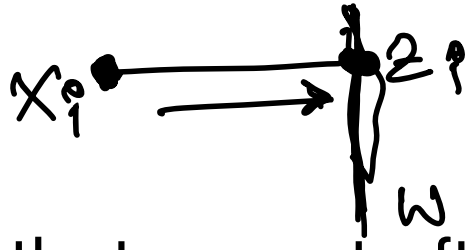


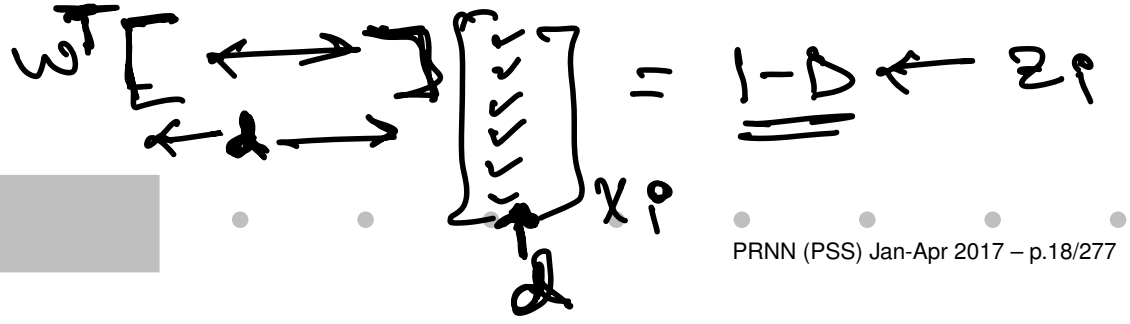
Fisher Linear Discriminant

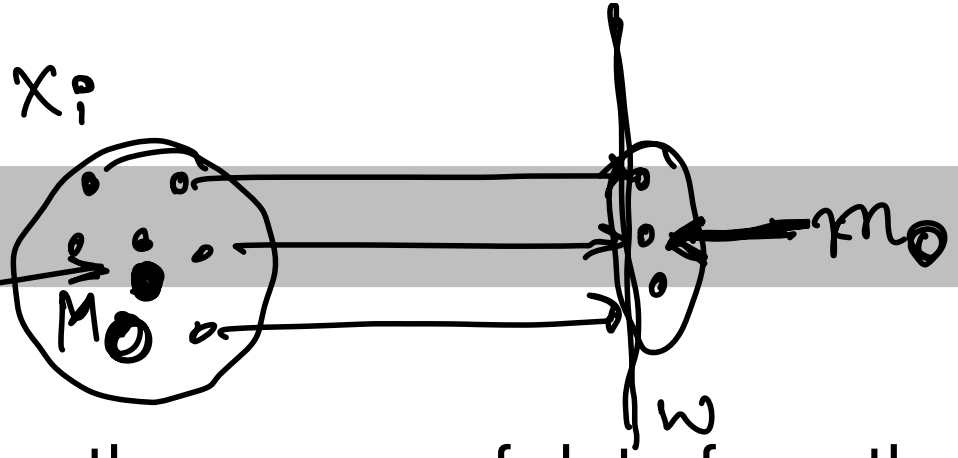
- The idea is to find a direction W such that the training data of the two classes are well-separated if projected onto this direction.
- We need some figure of merit for each W to characterize how well the W results in such a separation.
- We consider the 2-class case.

training data

- Let $\{(X_i,\ y_i),\ i = 1, \cdots, n\}$ be the data.
- Let $y_i \in \{0,\ 1\}$.
- Let C_0 and C_1 denote the two classes. Thus, if $y_i = 0$ then $X_i \in C_0$ and if $y_i = 1$ then $X_i \in C_1$.
- Let n_0 and n_1 denote the number of examples of each class. $(n = n_0 + n_1)$
- For any W, let $z_i = W^T X_i$.
- z_i are the one dimensional data that we get after projection.







- Let M_0 and M_1 be the means of data from the two classes:

$$M_0 = \frac{1}{n_0} \sum_{X_i \in C_0} X_i; \quad M_1 = \frac{1}{n_1} \sum_{X_i \in C_1} X_i$$

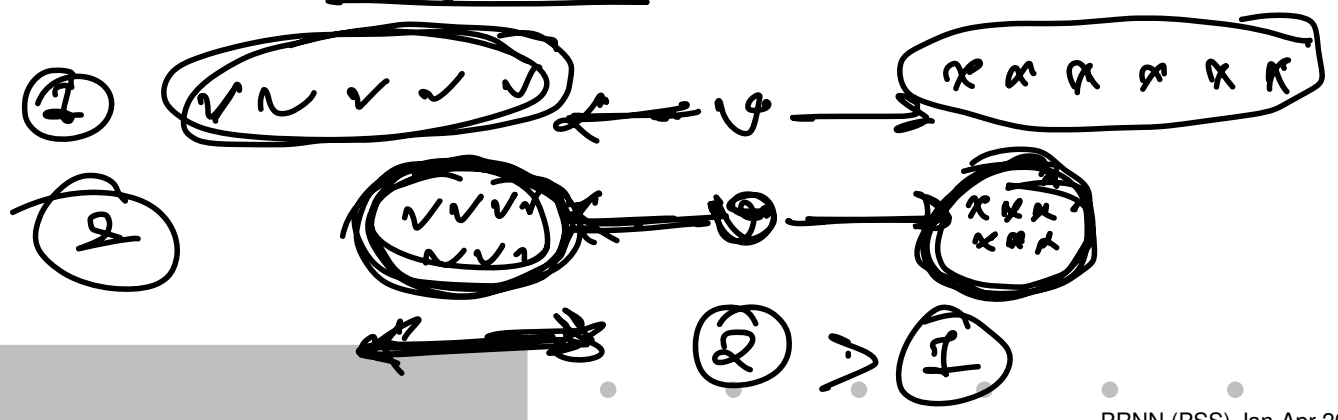
The corresponding means of the projected data would be

$$m_0 = W^T M_0 \quad \text{and} \quad m_1 = W^T M_1$$

$$\frac{1}{n_0} \sum_{x_i \in C_0} W^T x_i = \frac{1}{n_0} W^T \sum_{x_i \in C_0} x_i$$

$$= W^T M_0$$

- The difference $(m_0 - m_1)$ gives us an idea of the separation between samples of the two classes after projecting the data onto the direction W.
- Hence, we may want a W that maximizes $(m_0 - m_1)^2$.
- However, we have to make this scale independent.
- Also, the distance between means should be viewed relative to the variances.



- Define

$$s_0^2 = \sum_{X_i \in C_0} (W^T X_i - m_0)^2; \quad s_1^2 = \sum_{X_i \in C_1} (W^T X_i - m_1)^2$$

  These give us the variances (upto a factor) of the two classes in the projected data.

- We want large separation between m_0 and m_1 relative to the variances.

- Hence we can take our objective to be to maximize

$$J(W) = \frac{(m_1 - m_0)^2}{s_0^2 + s_1^2}$$

- We now rewrite J into a more convenient form.
- We have

$$\begin{aligned}(m_1 - m_0)^2 &= (W^T M_1 - W^T M_0)^2 \\ &= W^T (M_1 - M_0)(M_1 - M_0)^T W\end{aligned}$$

$$x^2 = x^T x$$

$$\left[W^T (M_1 - M_0)\right]^2$$

$$S_B$$

$v^T (M_1 - M_0)(M_1 - M_0)^T v = \|v^T (M_1 - M_0)\|^2$

$v^T S_B v \geq 0 \quad \forall v \in \mathbb{R}^d$

≥ 0

$x_i \to \mathbb{R}^d$

$M_1, M_0 \to \mathbb{R}^d$

- Thus we have $(m_1 - m_0)^2 = W^T S_B W$ where

Symmetric, P.S.D., Rank 1

$$S_B = (M_1 - M_0)(M_1 - M_0)^T.$$

- Here, S_B is a $d \times d$ matrix (note that $X_i \in \Re^d$).
- It is called *between class* scatter matrix.
- We can similarly write s_0^2 and s_1^2 also as quadratic forms.

We have

$$
\begin{aligned}
s_0^2 &= \sum_{X_i \in C_0} (W^T X_i - W^T M_0)^2 \\
&= \sum_{X_i \in C_0} [W^T (X_i - M_0)]^2 \\
&= \sum_{X_i \in C_0} W^T (X_i - M_0)(X_i - M_0)^T W \\
&= W^T \left[\sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T \right] W
\end{aligned}
$$

- Similarly, we get

$$s_1^2 = W^T \left[\sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T \right] W$$

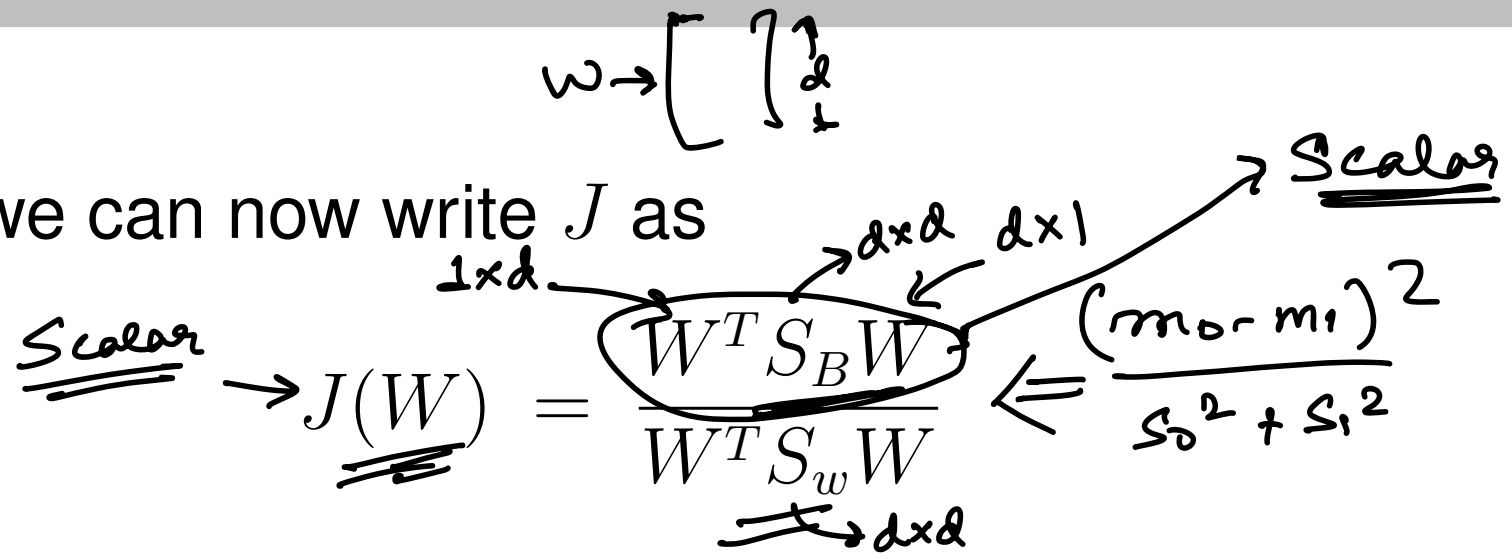
- Thus we can write $s_0^2 + s_1^2 = W^T S_w W$, where

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- S_w is also $d \times d$ matrix and is called *within class* scatter matrix.

$\|S_w\| \neq 0$

- Hence we can now write J as



- We want to find a W that maximizes $J(W)$.
- Note that $J(W)$ is not affected by scaling of W.
- Given the data we can calculate the S_B and S_w.
- Maximizing ratio of quadratic forms is a standard optimization problem.

- We need to maximize

$$J(W) = \frac{W^T S_B W}{W^T S_w W}$$

- Differentiating w.r.t. W and equating to zero, we get

$$\left[\frac{2S_B W}{W^T S_w W} - \frac{W^T S_B W}{(W^T S_w W)^2} 2S_w W\right] = 0$$

- Implies, $S_B W$ is in the same direction as $S_w W$.

$S_B w - J(w) S_w \quad w \in D$

- Thus, any maximizer of $J(W)$ has to satisfy

$$S_w W = \lambda S_B W$$

for some constant λ.

- This is known as the generalized eigen value problem.
- There are standard methods to solve this problem using, e.g., LU decomposition.
- By solving the generalized eigen value problem we can find the best direction W.

- Often, the real symmetric matrix S_w would be invertible.
- Recall that

$$S_w = \sum_{X_i \in C_0} (X_i - M_0)(X_i - M_0)^T + \sum_{X_i \in C_1} (X_i - M_1)(X_i - M_1)^T$$

- This is a sum of large number of rank 1 matrices.

- If S_w is invertible, then we can write

$$S_w^{-1} S_B W = \lambda W$$

- We have

$$S_B W = (M_1 - M_0)(M_1 - M_0)^T W = k(M_1 - M_0)$$

where k is some constant. (note $k = (m_1 - m_0)$)

- Now we get (since scale factor in W is not relevant)

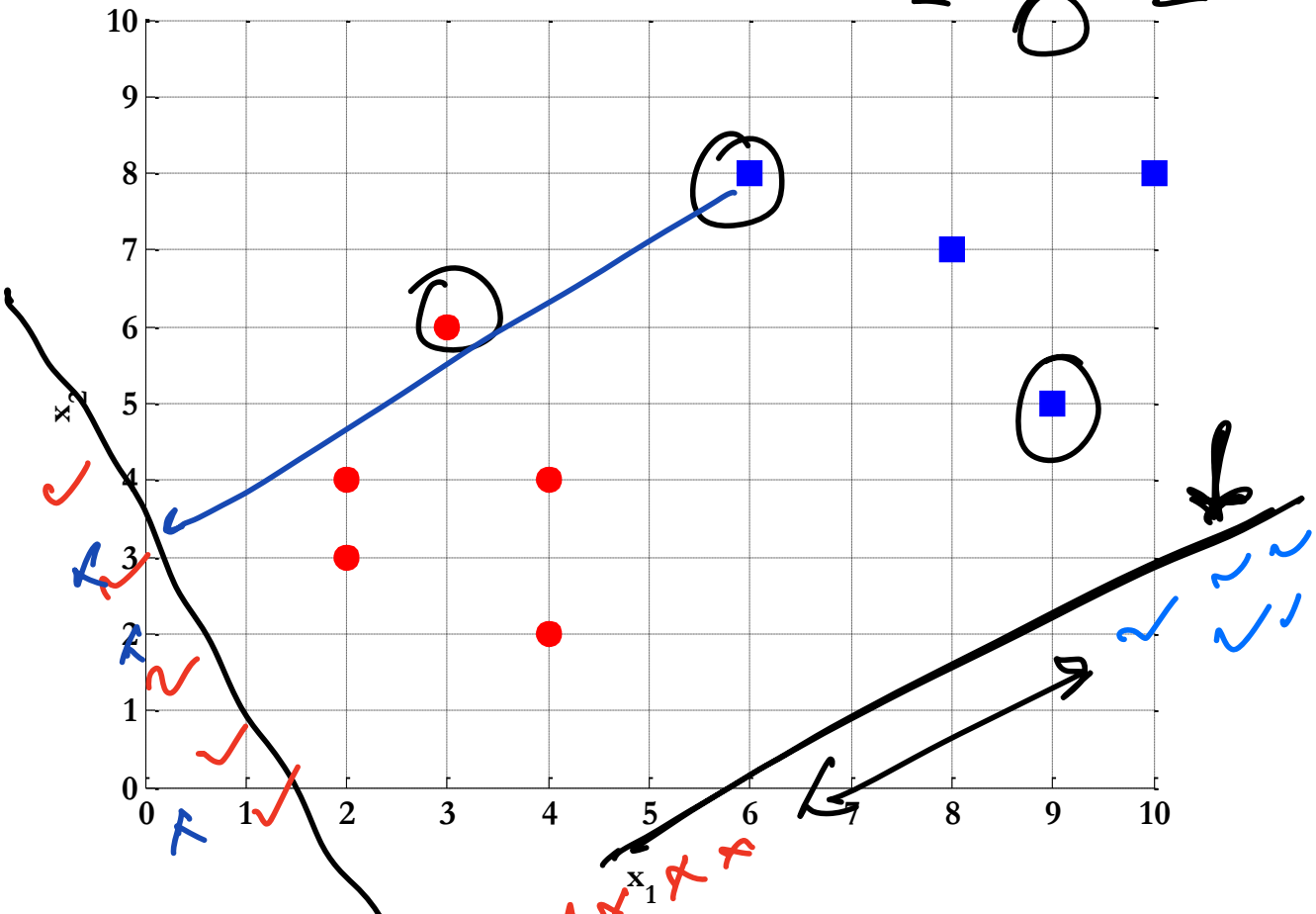
$$W = S_w^{-1}(M_1 - M_0)$$

Obtaining Fisher Linear Discriminant

- We can sum-up the process as follows.
- Given the training data, we first form the scatter matrix S_w and also calculate the means M_0 and M_1.
- If S_w is invertible, we calculate W by $W = S_w^{-1}(M_1 - M_0)$.
- Even if S_w is not invertible, there are techniques to find the maximizer of $J(W)$ by solving the generalized eigen value problem.
- Thus we can find the best direction W.

LDA … Two Classes - Example

- Compute the Linear Discriminant projection for the following two-dimensional dataset.
  - Samples for class ω_1 : $\mathbf{X}_1=(x_1,x_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$
  - Sample for class ω_2 : $\mathbf{X}_2=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$




```
% samples for class 1
X1 = [4,2;
      2,4;
      2,3;
      3,6;
      4,4];


% samples for class 2
X2 = [9,10;
      6,8;
      9,5;
      8,7;
      10,8];
```

LDA … Two Classes - Example

- The classes mean are :

$$\mu_1 = \frac{1}{N_1}\sum_{x\in\omega_1} x = \frac{1}{5}\left[\begin{pmatrix}4\\2\end{pmatrix}+\begin{pmatrix}2\\4\end{pmatrix}+\begin{pmatrix}2\\3\end{pmatrix}+\begin{pmatrix}3\\6\end{pmatrix}+\begin{pmatrix}4\\4\end{pmatrix}\right] = \begin{pmatrix}3\\3.8\end{pmatrix}$$

$$\mu_2 = \frac{1}{N_2}\sum_{x\in\omega_2} x = \frac{1}{5}\left[\begin{pmatrix}9\\10\end{pmatrix}+\begin{pmatrix}6\\8\end{pmatrix}+\begin{pmatrix}9\\5\end{pmatrix}+\begin{pmatrix}8\\7\end{pmatrix}+\begin{pmatrix}10\\8\end{pmatrix}\right] = \begin{pmatrix}8.4\\7.6\end{pmatrix}$$

```
% class means
Mu1 = mean(X1)';
Mu2 = mean(X2)';
```

LDA … Two Classes - Example

- Covariance matrix of the first class:

$$S_1 = \sum_{x \in \omega_1} (x - \mu_1)(x - \mu_1)^T = \left[\begin{pmatrix} 4 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 2 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2$$

$$+ \left[\begin{pmatrix} 2 \\ 3 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 3 \\ 6 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 4 \\ 4 \end{pmatrix} - \begin{pmatrix} 3 \\ 3.8 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix}$$

d=2

← intra class scatter matrix

```
% covariance matrix of the first class
S1 = cov(X1);
```

LDA … Two Classes - Example

- Covariance matrix of the second class:

$$S_2 = \sum_{x \in \omega_2} (x - \mu_2)(x - \mu_2)^T = \left[\begin{pmatrix} 9 \\ 10 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 6 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$+ \left[\begin{pmatrix} 9 \\ 5 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 8 \\ 7 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2 + \left[\begin{pmatrix} 10 \\ 8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^2$$

$$= \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

Intra Scatter

```
% covariance matrix of the first class
S2 = cov(X2);
```

LDA … Two Classes - Example

- Within-class scatter matrix:

$$S_w = S_1 + S_2 = \begin{pmatrix} 1 & -0.25 \\ -0.25 & 2.2 \end{pmatrix} + \begin{pmatrix} 2.3 & -0.05 \\ -0.05 & 3.3 \end{pmatrix}$$

$$= \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}$$

$S_w^{-1}(M_1 - M_0)$

```
% within-class scatter matrix
Sw = S1 + S2 ;
```

LDA … Two Classes - Example

- Between-class scatter matrix:

$S_B^{-1} S_B = 1$

$\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$

$$S_B = (\mu_1 - \mu_0)(\mu_1 - \mu_0)^T$$

$$= \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]\left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix}\right]^T$$

$$= \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}\begin{pmatrix} -5.4 & -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix}$$

Inter class Scatter

```
% between-class scatter matrix
SB = (Mu1-Mu2)*(Mu1-Mu2)';
```

LDA … Two Classes - Example

- The LDA projection is then obtained as the solution of the generalized eigen value problem

$$S_W^{-1} S_B w = \lambda w \rightarrow S_W^{-1} S_B w - \lambda w = 0$$

$$\Rightarrow \left| S_W^{-1} S_B - \lambda I \right| = 0 \qquad \left(S_W^{-1} S_B - \lambda I\right) w = 0$$

$$\Rightarrow \left| \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} 29.16 & 20.52 \\ 20.52 & 14.44 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right| = 0$$

$$\Rightarrow \left| \begin{pmatrix} 9.2213 - \lambda & 6.489 \\ 4.2339 & 2.9794 - \lambda \end{pmatrix} \right| = 0$$

$$= (9.2213 - \lambda)(2.9794 - \lambda) - 6.489 \times 4.2339 = 0$$

$$\Rightarrow \lambda^2 - 12.2007\lambda = 0 \Rightarrow \lambda(\lambda - 12.2007) = 0$$

$$\Rightarrow \lambda_1 = 0, \lambda_2 = 12.2007$$

$Ax = \lambda x$

$|A - \lambda I| = 0$

$A = S_W^{-1} S_B$

A

LDA … Two Classes - Example

```
% computing the LDA projection
invSw = inv(Sw);

invSw_by_SB = invSw * SB;

% getting the projection vector
[V,D] = eig(invSw_by_SB)

% the projection vector
W = V(:,1);
```

- Hence

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_1 = \underbrace{0}_{\lambda_1} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

and

$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_2 = \underbrace{12.2007}_{\lambda_2} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

Thus;

$$w_1 = \begin{pmatrix} -0.5755 \\ 0.8178 \end{pmatrix} \quad and \quad w_2 = \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} = w^*$$

- The optimal projection is the one that given maximum $\lambda = J(w)$

LDA … Two Classes - Example

Or directly;

S_W^{-1} is invertible

$$w^* = S_W^{-1}(\mu_1 - \mu_2) = \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]$$

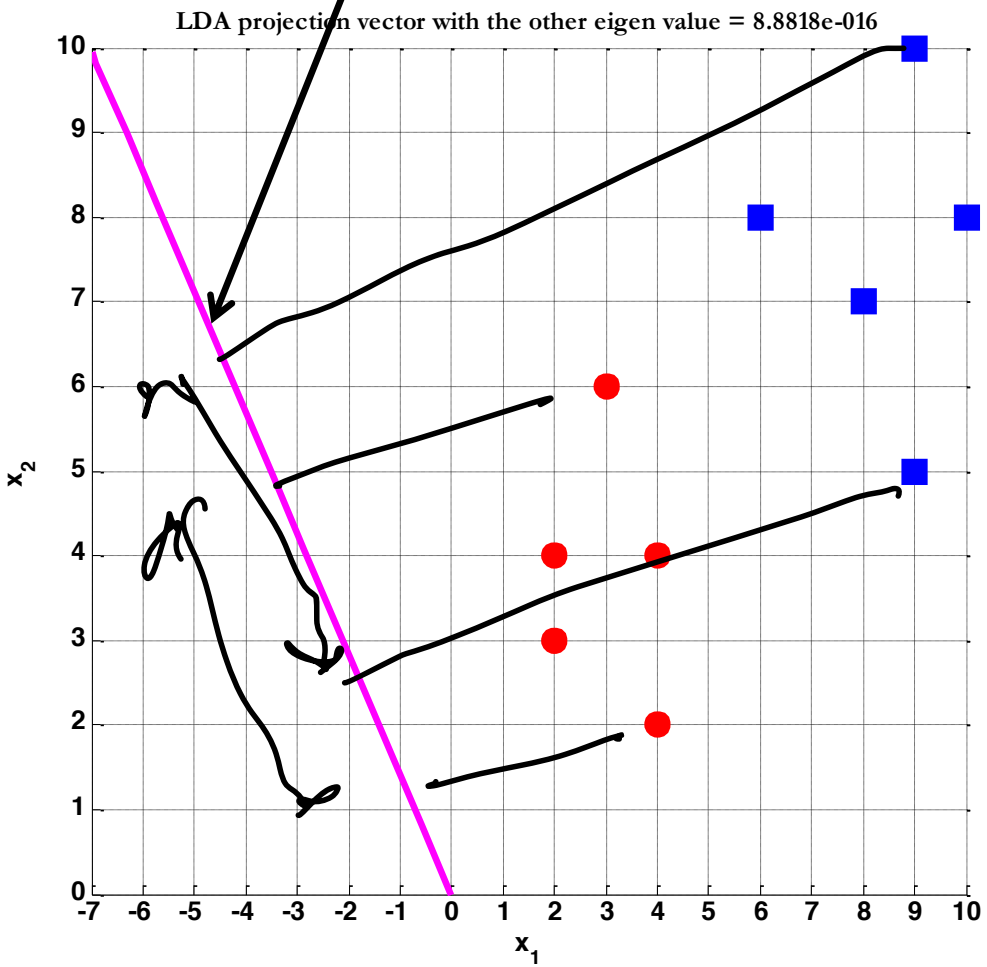
$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

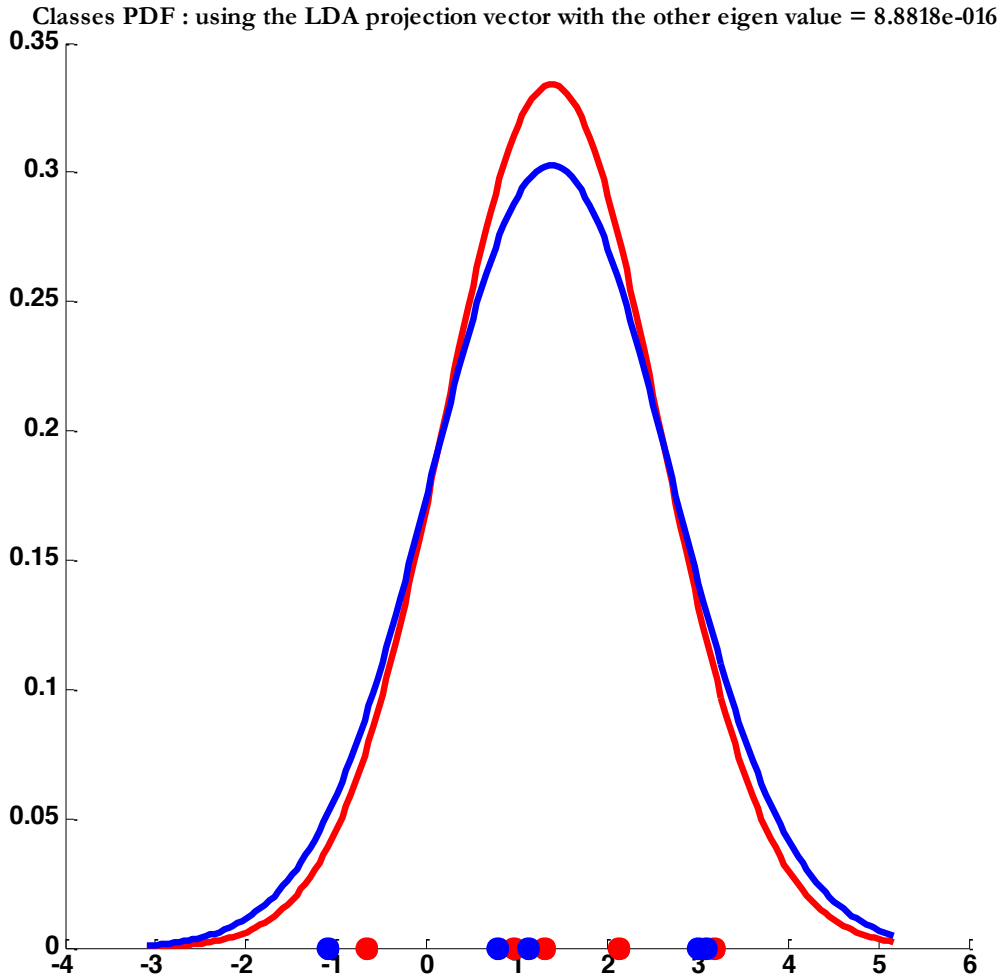
$$= \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix} \leftarrow w^*$$

LDA - Projection

The projection vector corresponding to the **smallest** eigen value

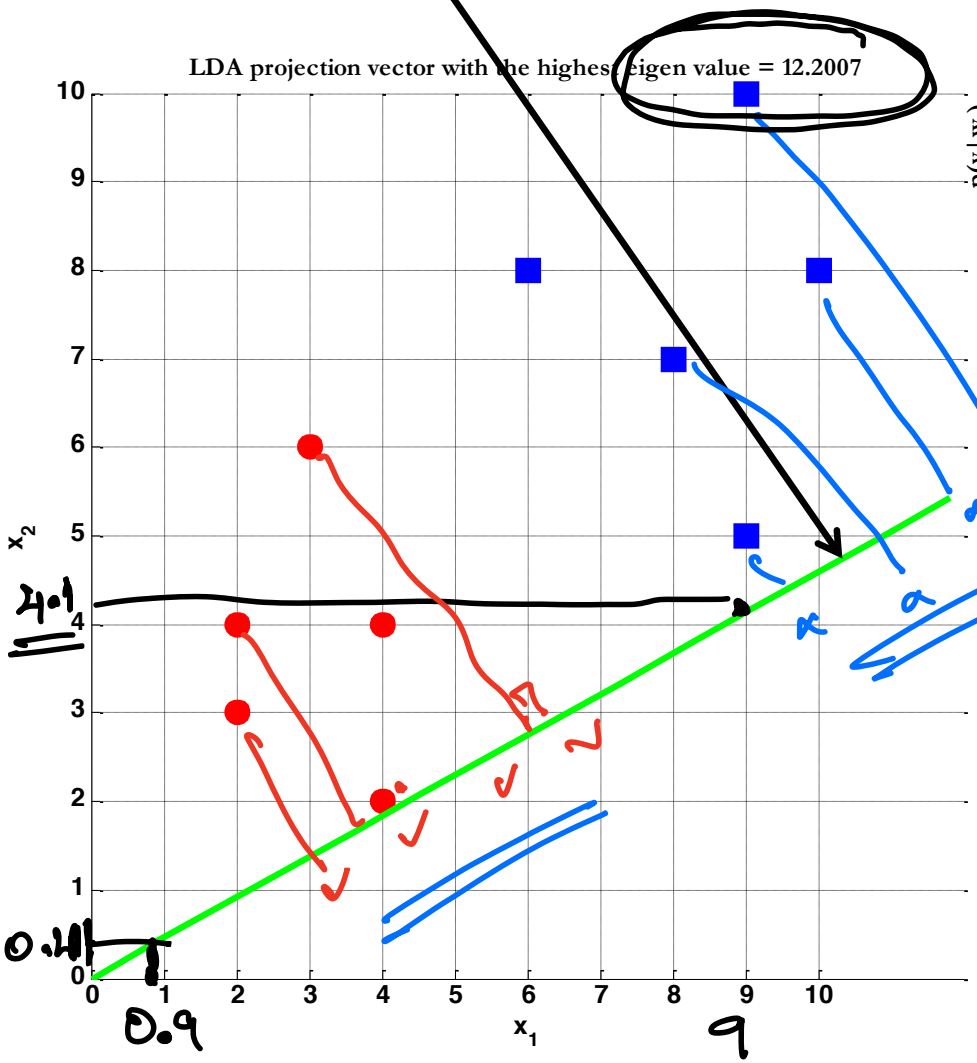


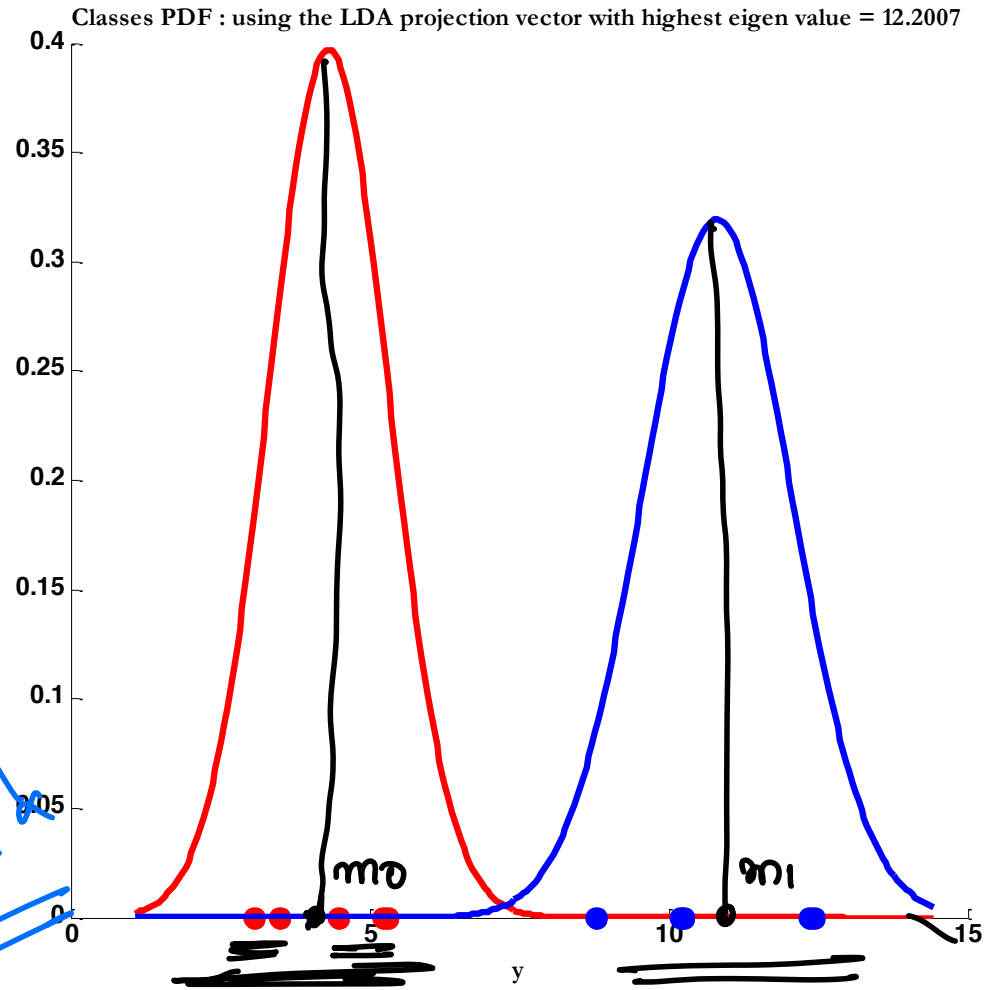


Using this vector leads to **bad separability** between the two classes

LDA - Projection

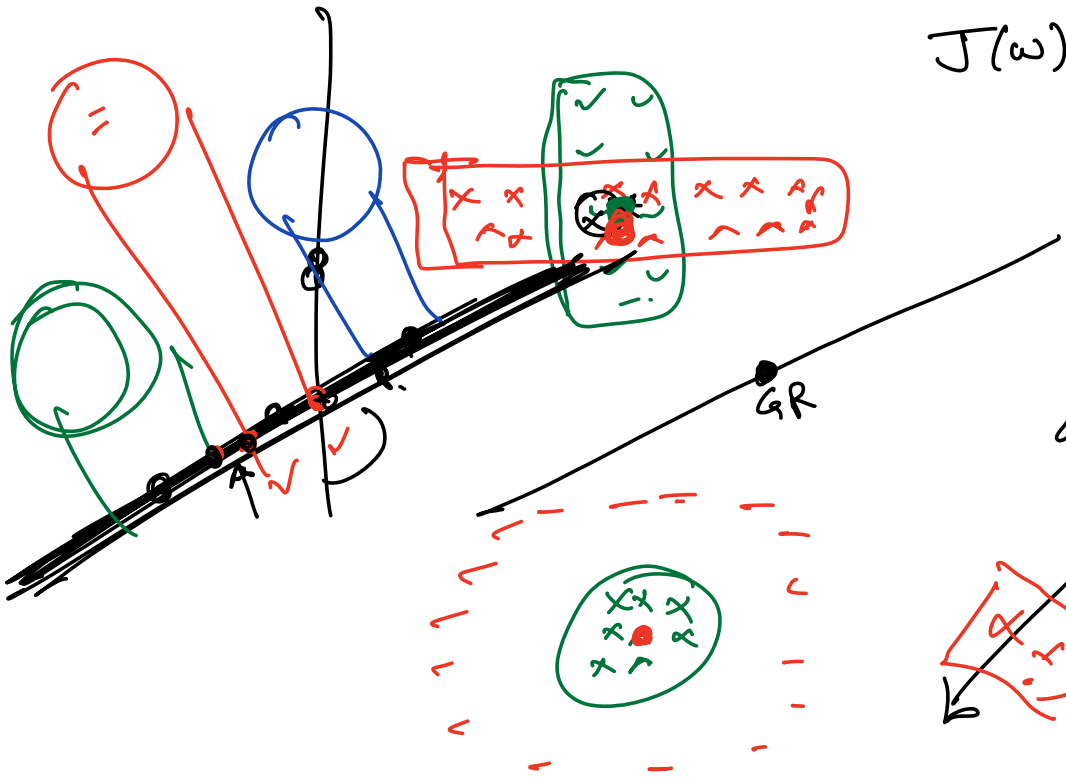
The projection vector corresponding to the **highest** eigen value

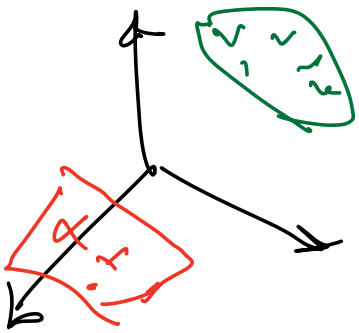




Using this vector leads to **good separability** between the two classes

$$J(\omega) = \frac{(m_1 - m_0)^2}{S_0^2 + S_1^2}$$





LDA … C-Classes

- Now, we have *C*-classes instead of just two.
- We are now seeking (*C-1*) projections $[\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_{C-1}}]$ by means of (*C-1*) projection vectors $\mathbf{w_i}$.
- $\mathbf{w_i}$ can be arranged by *columns* into a projection matrix $\mathbf{W} = [\mathbf{w_1}|\mathbf{w_2}|\ldots|\mathbf{w_{C-1}}]$ such that:

$$y_i = w_i^T x \quad \Rightarrow \quad y = W^T x$$

projection

$$where \quad x_{m\times 1} = \begin{bmatrix} x_1 \\ . \\ . \\ x_m \end{bmatrix}, \quad y_{C-1\times 1} = \begin{bmatrix} y_1 \\ . \\ . \\ y_{C-1} \end{bmatrix}$$

$$and \quad W_{m\times C-1} = \begin{bmatrix} w_1 | & w_2 | & \ldots & | w_{C-1} \end{bmatrix}$$

LDA … C-Classes

- If we have *n*-feature vectors, we can stack them into one matrix as follows;

$$Y = W^T X$$

feature matrix

$$where \quad X_{m\times n} = \begin{bmatrix} x_1^1 & x_1^2 & . & x_1^n \\ . & . & . & . \\ . & . & . & . \\ x_m^1 & x_m^2 & . & x_m^n \end{bmatrix} \quad , \quad Y_{C-1\times n} = \begin{bmatrix} y_1^1 & y_1^2 & . & y_1^n \\ . & . & . & . \\ . & . & . & . \\ y_{C-1}^1 & y_{C-1}^2 & . & y_{C-1}^n \end{bmatrix}$$

C-1 dim

$$and \quad W_{m\times C-1} = [w_1 \mid \; w_2 \mid \; \ldots \; \mid w_{C-1}]$$

LDA – C-Classes

- Recall the two classes case, the *within-class scatter* was computed as:

$$S_w = S_1 + S_2$$

- This can be generalized in the *C*-classes case as:

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

where

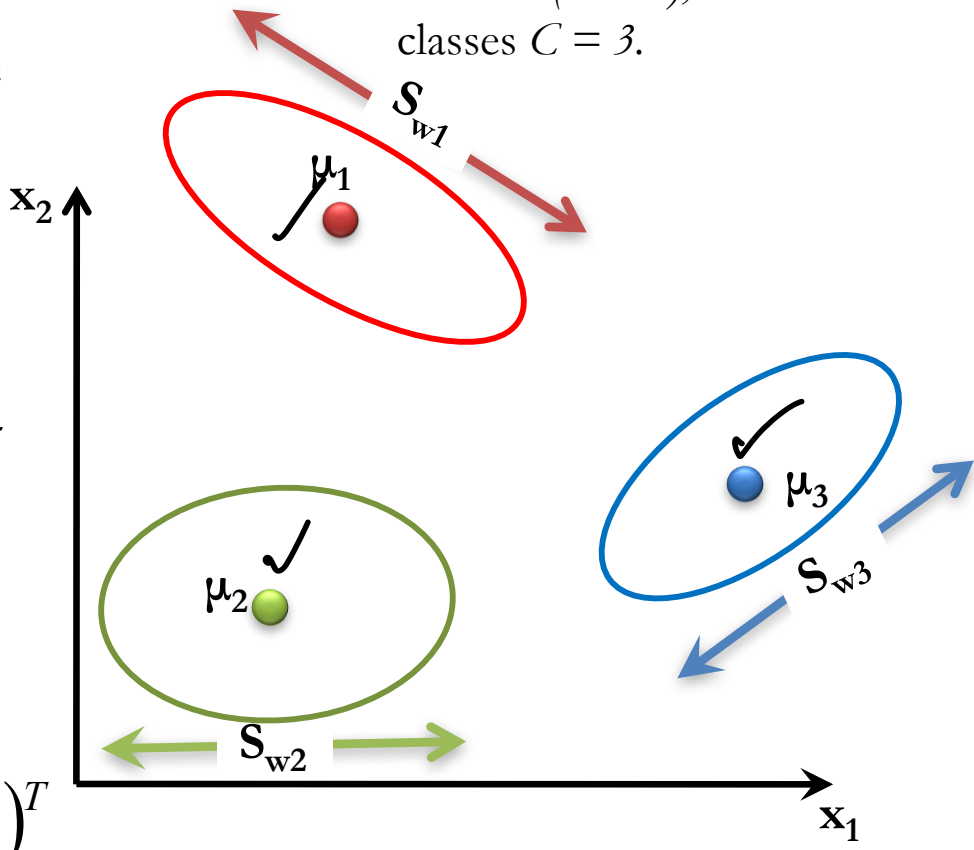
$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

and

$$\mu_i = \frac{1}{N_i} \sum_{x \in \omega_i} x$$

$\mathbf{N_i}$: number of data samples in class $\boldsymbol{\omega_i}$.

Example of two-dimensional features *(m = 2)*, with three classes *C = 3*.



LDA – C-Classes

- Recall the two classes case, the *between-class scatter* was computed as:

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

- For *C*-classes case, we will measure the between-class scatter with respect to the mean of all classes as follows:

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

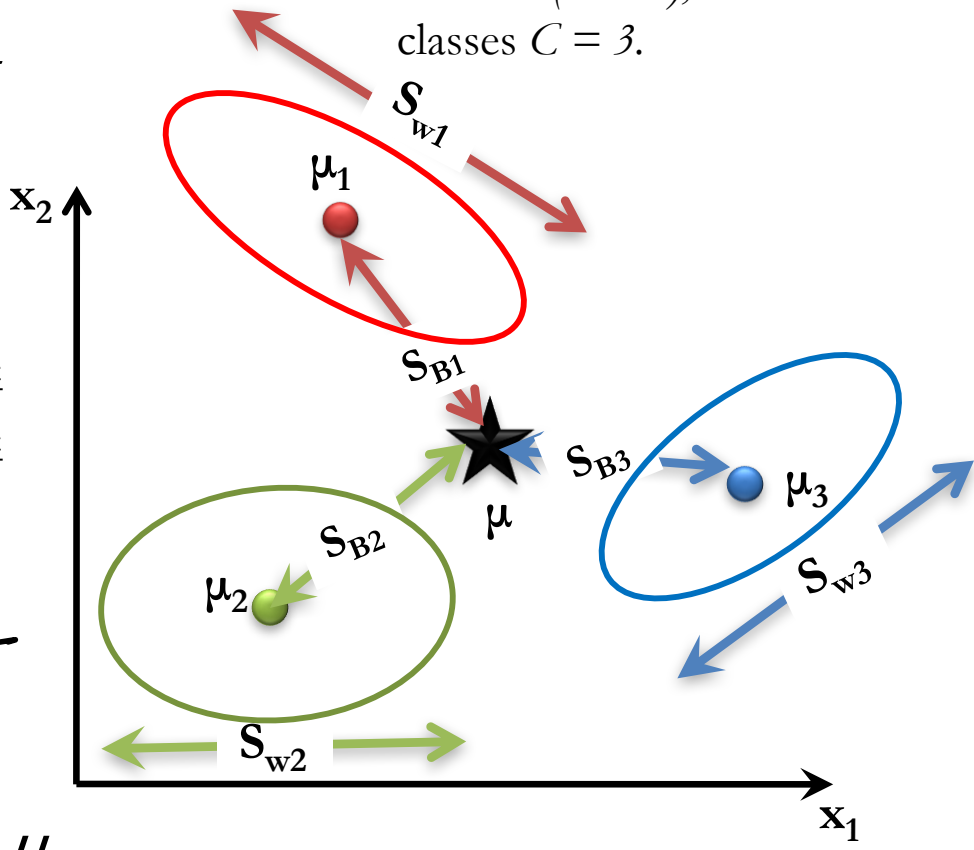
where $$\mu = \frac{1}{N}\sum_{\forall x} x = \frac{1}{N}\sum_{\forall x} N_i \mu_i$$

N: number of all data .

and $$\mu_i = \frac{1}{N_i}\sum_{x \in \omega_i} x$$

$\mathbf{N_i}$: number of data samples in class $\boldsymbol{\omega_i}$.

Example of two-dimensional features *(m = 2)*, with three classes *C = 3*.



LDA – C-Classes

- Recall in two-classes case, we have expressed the scatter matrices of the projected samples in terms of those of the original samples as:

$$\Rightarrow \tilde{S}_W = W^T S_W W$$

$$\Rightarrow \tilde{S}_B = W^T S_B W$$

This still hold in *C*-classes case.

- Recall that we are looking for a projection that maximizes the ratio of between-class to within-class scatter.

- Since the projection is no longer a scalar (it has *C-1* dimensions), we then use the determinant of the scatter matrices to obtain a scalar objective function:

$$J(W) = \frac{\left|\tilde{S}_B\right|}{\left|\tilde{S}_W\right|} = \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|}$$

$1 \times d \quad d \times d \quad d \times 1$

Scalar

$C-1 \times d \quad d \times d \quad d \times C-1$

- And we will seek the projection $\mathbf{W}^*$ that maximizes this ratio.

LDA – C-Classes

- To find the maximum of *J(W)*, we differentiate with respect to **W** and equate to zero.

- Recall in two-classes case, we solved the eigen value problem.

$$S_W^{-1} S_B w = \lambda w \qquad where \quad \lambda = J(w) = scalar$$

- For *C*-classes case, we have *C-1* projection vectors, hence the eigen value problem can be generalized to the *C*-classes case as:

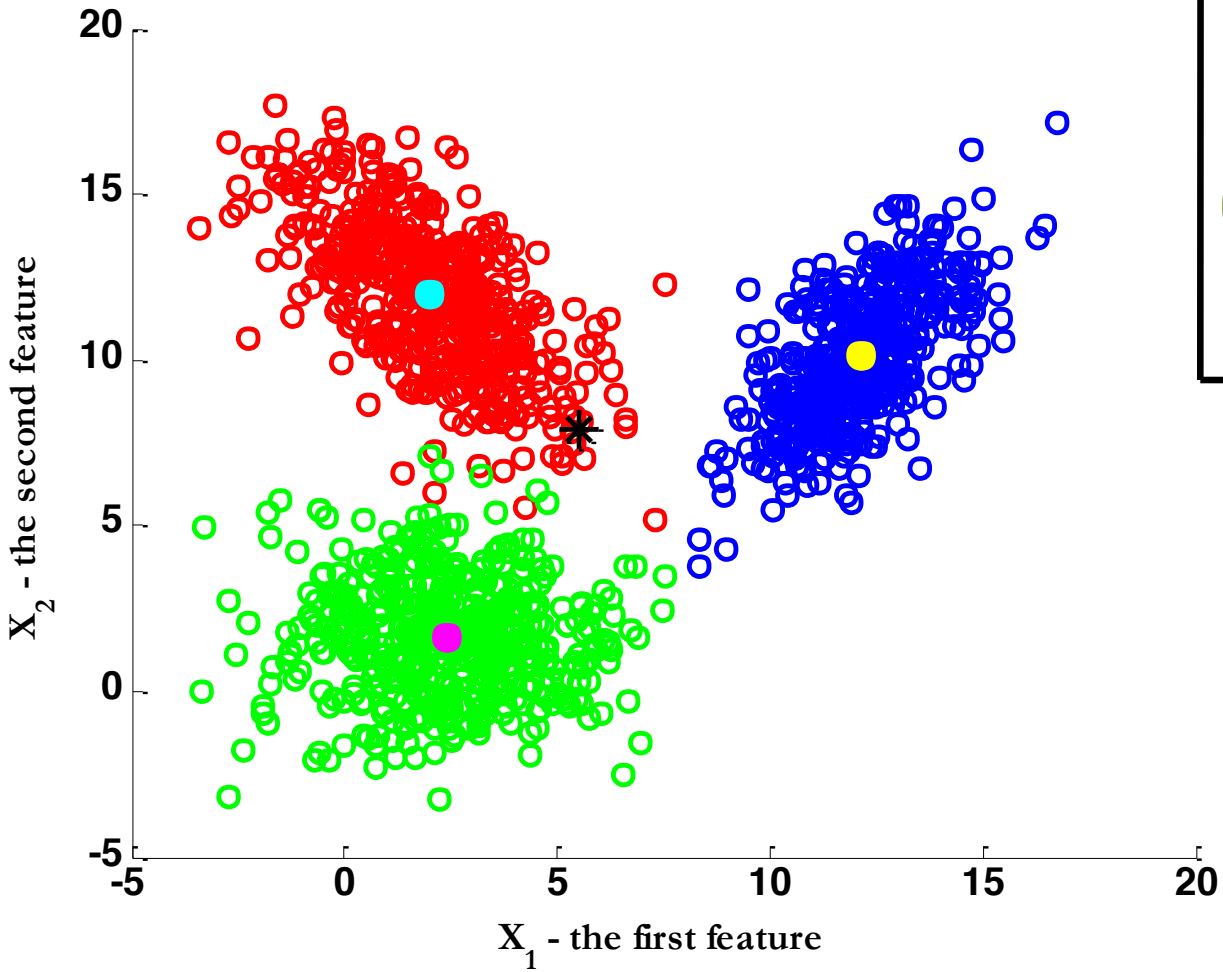
$$S_W^{-1} S_B w_i = \lambda_i w_i \qquad where \quad \lambda_i = J(w_i) = scalar \qquad and \quad i = 1,2,...C-1$$

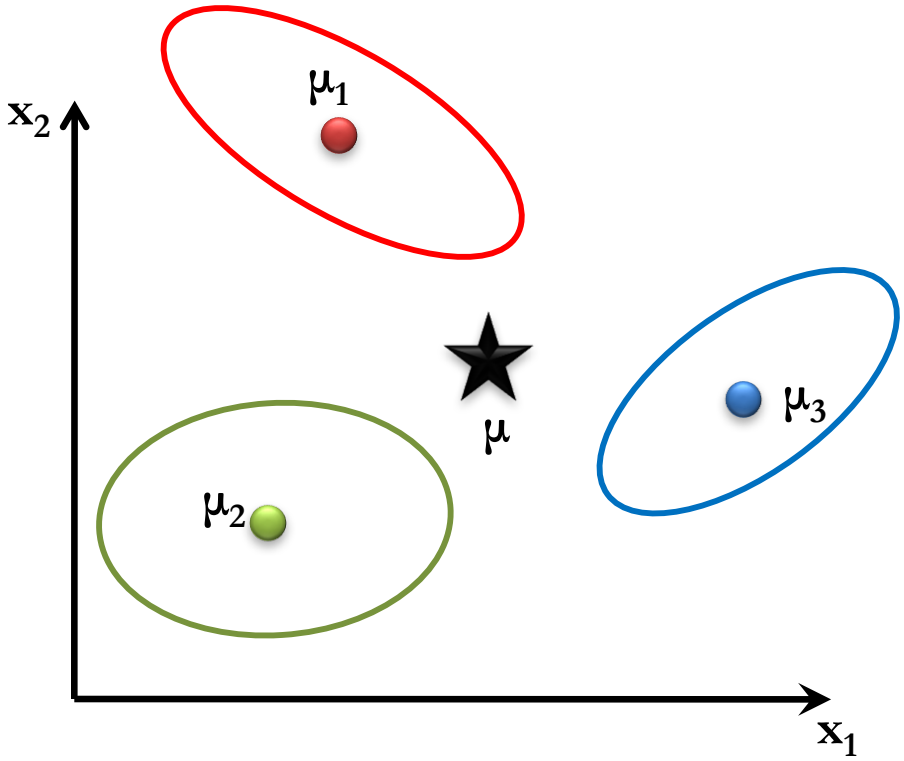
- Thus, It can be shown that the optimal projection matrix $\mathbf{W}^*$ is the one whose columns are the eigenvectors corresponding to the largest eigen values of the following generalized eigen value problem:

$$S_W^{-1} S_B W^* = \lambda W^*$$

$$where \quad \lambda = J(W^*) = scalar \qquad and \qquad W^* = \left[w_1^* \mid \; w_2^* \mid \; ... \; \mid w_{C-1}^* \right]$$

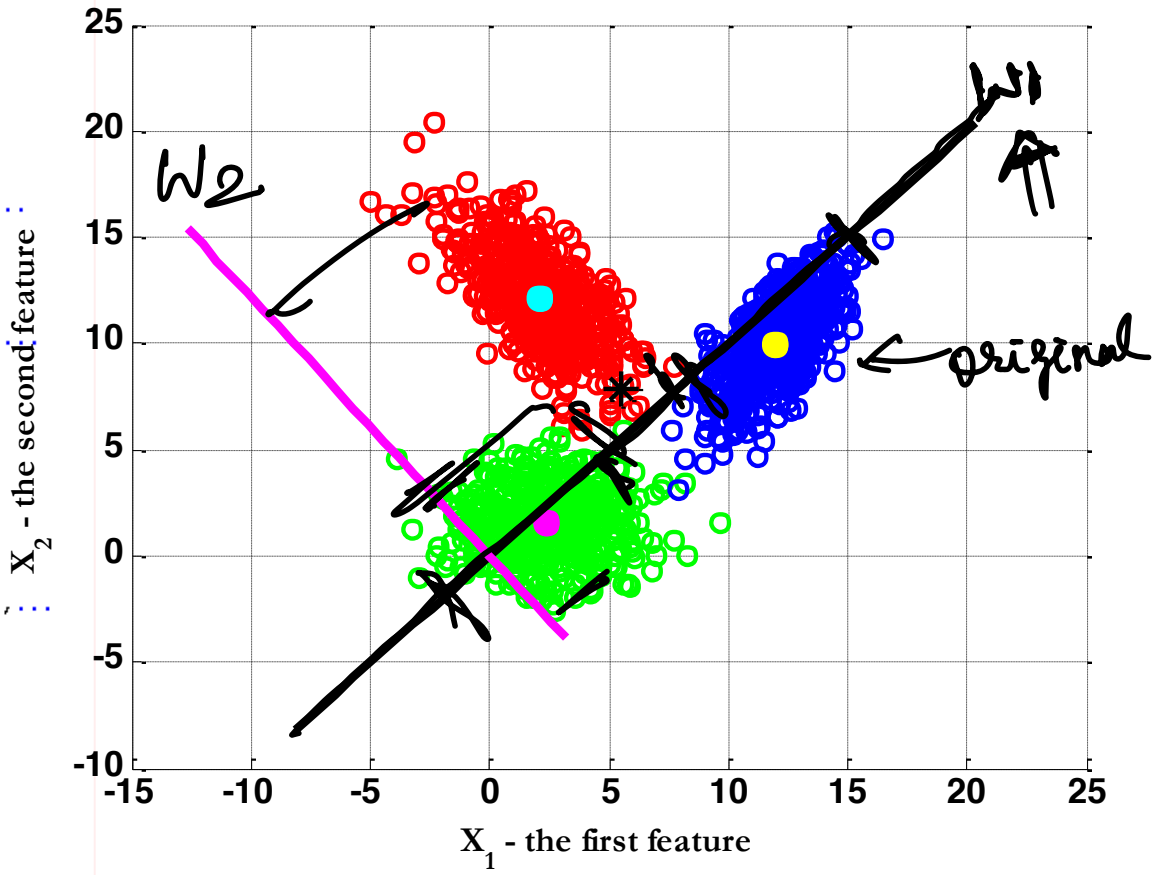
It's Working … ☺


20
15
10
5
0
-5
-5
0
5
10
15
20
X2 - the second feature
X1 - the first feature


x2
x1
μ1
μ2
μ3
μ

Let's visualize the projection vectors *W*

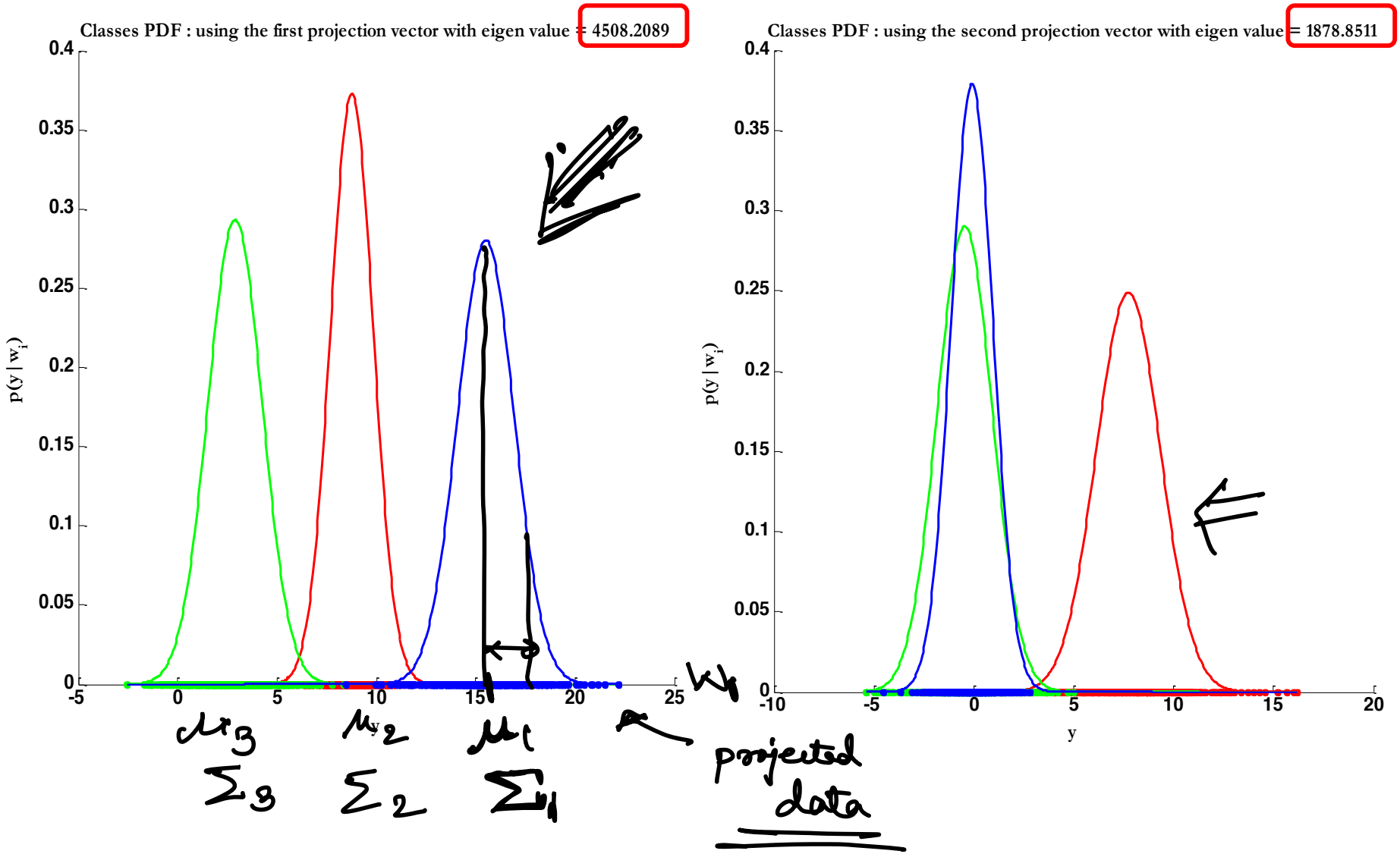
```
%% lets visualize them ...
% we will plot the scatter plot to better visualize the features
hfig = figure;
axes1 = axes('Parent',hfig,'FontWeight','bold','FontSize',12);
hold('all');

% Create xlabel
xlabel('X_1 - the first feature','FontWeight','bold','FontSize',12,...
    'FontName','Garamond');

% Create ylabel
ylabel('X_2 - the second feature','FontWeight','bold','FontSize',12,...
    'FontName','Garamond');

% the first class
scatter(X1(1,:),X1(2,:), 'r','LineWidth',2,'Parent',axes1);
hold on

% class's mean
plot(Mu1_est(1),Mu1_est(2),'co','MarkerSize',8,'MarkerEdgeColor','c',...
    'Color','c','LineWidth',2,'MarkerFaceColor','c','Parent',axes1);
hold on

% the second class
scatter(X2(1,:),X2(2,:), 'g','LineWidth',2,'Parent',axes1);
hold on

% class's mean
plot(Mu2_est(1),Mu2_est(2),'mo','MarkerSize',8,'MarkerEdgeColor','m',...
    'Color','m','LineWidth',2,'MarkerFaceColor','m','Parent',axes1);
hold on

% the third class
scatter(X3(1,:),X3(2,:), 'b','LineWidth',2,'Parent',axes1);
hold on

% class's mean
plot(Mu3_est(1),Mu3_est(2),'yo','LineWidth',2,'MarkerSize',8,'MarkerEdgeColor',...
    'y','Color','y','MarkerFaceColor','y','Parent',axes1);
hold on
```




```
% drawing the projection vectors
% the first vector
t = -10:25;
line_x1 = t .* W1(1);
line_y1 = t .* W1(1);

% the second vector
t = -5:20;
line_x2 = t .* W2(1);
line_y2 = t .* W2(2);

plot(line_x1,line_y1,'k-','LineWidth',3);
hold on
plot(line_x2,line_y2,'m-','LineWidth',3);
grid on
```

Which is Better?!!!

- Apparently, the projection vector that has the **highest eigen value** provides higher discrimination power between classes



$$\mathcal{N}(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right\}$$

Dimension = $c-1$

Any new test point

$C_1 \ldots C_K$

$$P(C_K \mid x) = p(x \mid C_K)\, P(C_K)$$

μ_K, Σ_K

prior

x_{test} → $P_{C_1} \mid x_{test}$, $P_{C_2} \mid x_{test}$, ⋮, $P_{C_K} \mid x_{test}$

argmax $\frac{\#\text{training points in class } K}{\#\text{training points}}$