PCA and LDA

Outline

- Statistics Review
- Principle Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- Compare PCA and LDA

Statistics Basics

- Given a data X: $X = [\,1\ 2\ 4\ 6\ 12\ 15\ 25\ 45\ 68\ 67\ 65\ 98\,]$
- Mean of the data: $\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$
- Standard deviation of the data: $s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}}$
- Variance of the data: $s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$
- Co-variance of the data:
  - For two dimensions, measures the dimensions vary from the mean w. r. t. each other $cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$
    - cov(x, y)>0 ➔increase x will see increase of y
    - cov(x, y)<0 ➔increase x will see decrease of y
    - cov(x, y)=0 ➔ x and y are independent

Statistics Basics

- Covariance Matrix: $C^{n\times n} = (c_{i,j},\ c_{i,j} = cov(Dim_i, Dim_j))$
  - For a data described with n dimensions, compute the covariance between each pair of dimensions
  - In the case where data is described with 3 dimensions:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

  - n x n square matrix
  - symmetric

Matrix Algebra

- Eigenvalues and Eigenvectors of a Matrix

$$A\,\vec{v} = \lambda\,\vec{v}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

- Compute the Eigen values:

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

A: matrix, I: identity matrix, λ: eigenvalues

An Example

$$\det\left(\begin{pmatrix}504 & 360 & 180\\ 360 & 360 & 0\\ 180 & 0 & 720\end{pmatrix} - \lambda\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\right)$$

$$= \det\begin{pmatrix}504 & 360 & 180\\ 360 & 360 & 0\\ 180 & 0 & 720\end{pmatrix} - \begin{pmatrix}\lambda & 0 & 0\\ 0 & \lambda & 0\\ 0 & 0 & \lambda\end{pmatrix}$$

$$= \det\begin{pmatrix}504-\lambda & 360 & 180\\ 360 & 360-\lambda & 0\\ 180 & 0 & 720-\lambda\end{pmatrix}$$

$$= -\lambda^3 + 1584\lambda^2 - 641520\lambda + 25660800$$

$$-\lambda^3 + 1584\lambda^2 - 641520\lambda + 25660800 = 0$$

Eigenvalues:

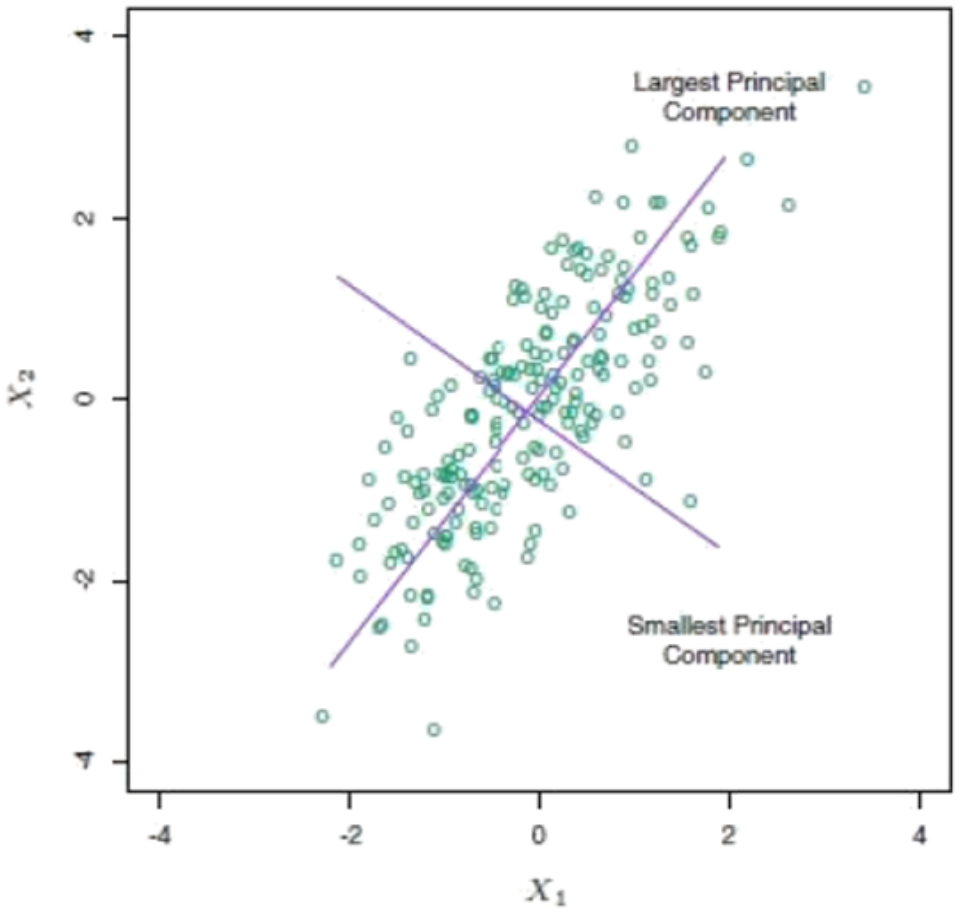
$$\lambda \approx 44.81966\ldots,\ \lambda \approx 629.11039\ldots,\ \lambda \approx 910.06995\ldots$$

Eigenvectors:

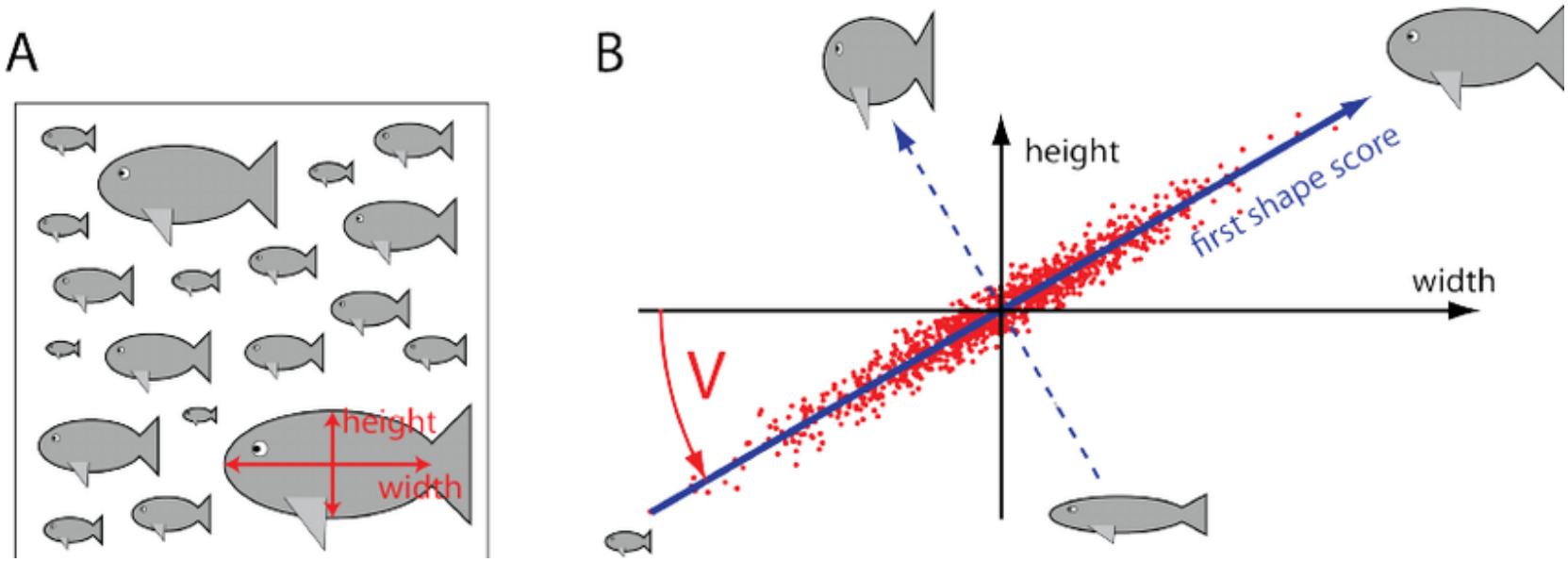
$$\begin{pmatrix}-3.75100\ldots\\ 4.28441\ldots\\ 1\end{pmatrix}, \begin{pmatrix}-0.50494\ldots\\ -0.67548\ldots\\ 1\end{pmatrix}, \begin{pmatrix}1.05594\ldots\\ 0.69108\ldots\\ 1\end{pmatrix}$$

PCA

- Reduce the dimensionality of a data set consisting of a large number of interrelated variables while retaining as much as possible of the variation present in the data set.



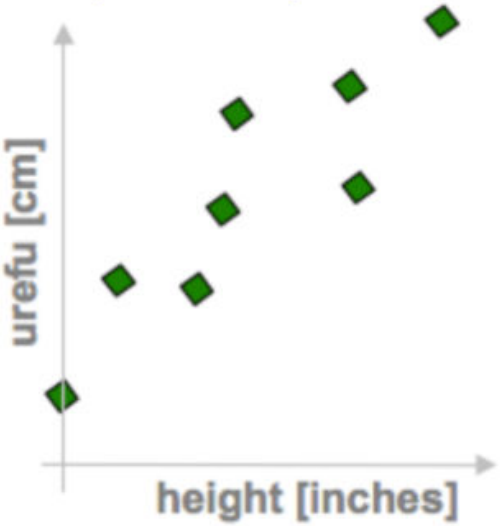
PCA Example



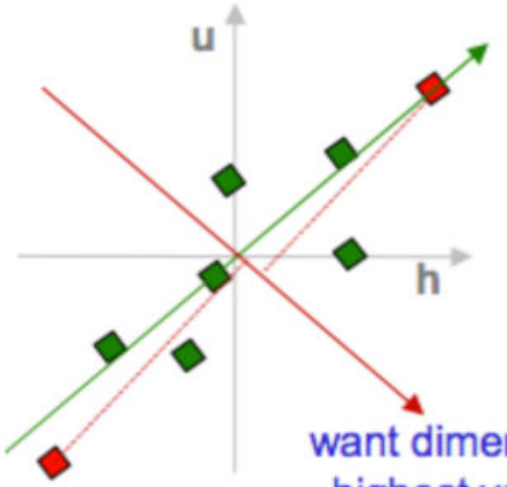
PCA in a nutshell

1. correlated hi-d data

("urefu" means "height" in Swahili)



2. center the points



want dimension of highest variance

3. compute covariance matrix

	h	u
h	2.0	0.8
u	0.8	0.6

$$\text{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$$

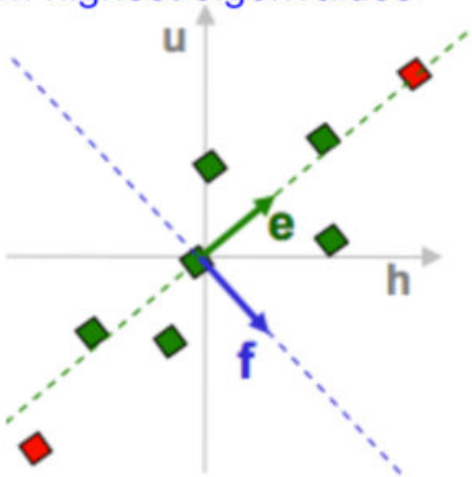
4. eigenvectors + eigenvalues

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$$

$$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix} \begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$$

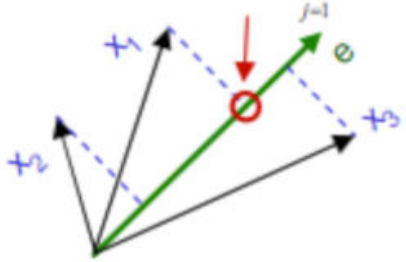
```
eig(cov(data))
```

5. pick m<d eigenvectors w. highest eigenvalues

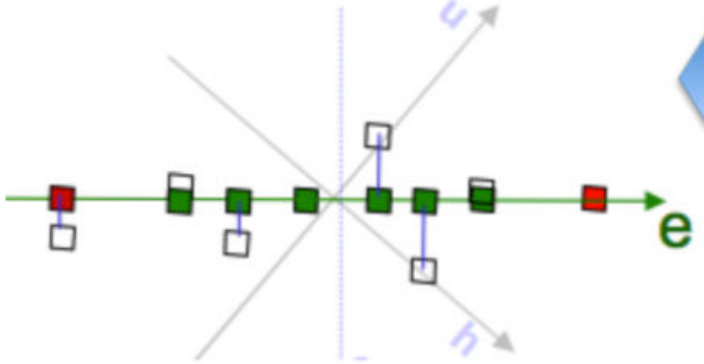


6. project data points to those eigenvectors

$$x_e' = x^T e = \sum_{j=1}^{a} x_{ij} e_j$$



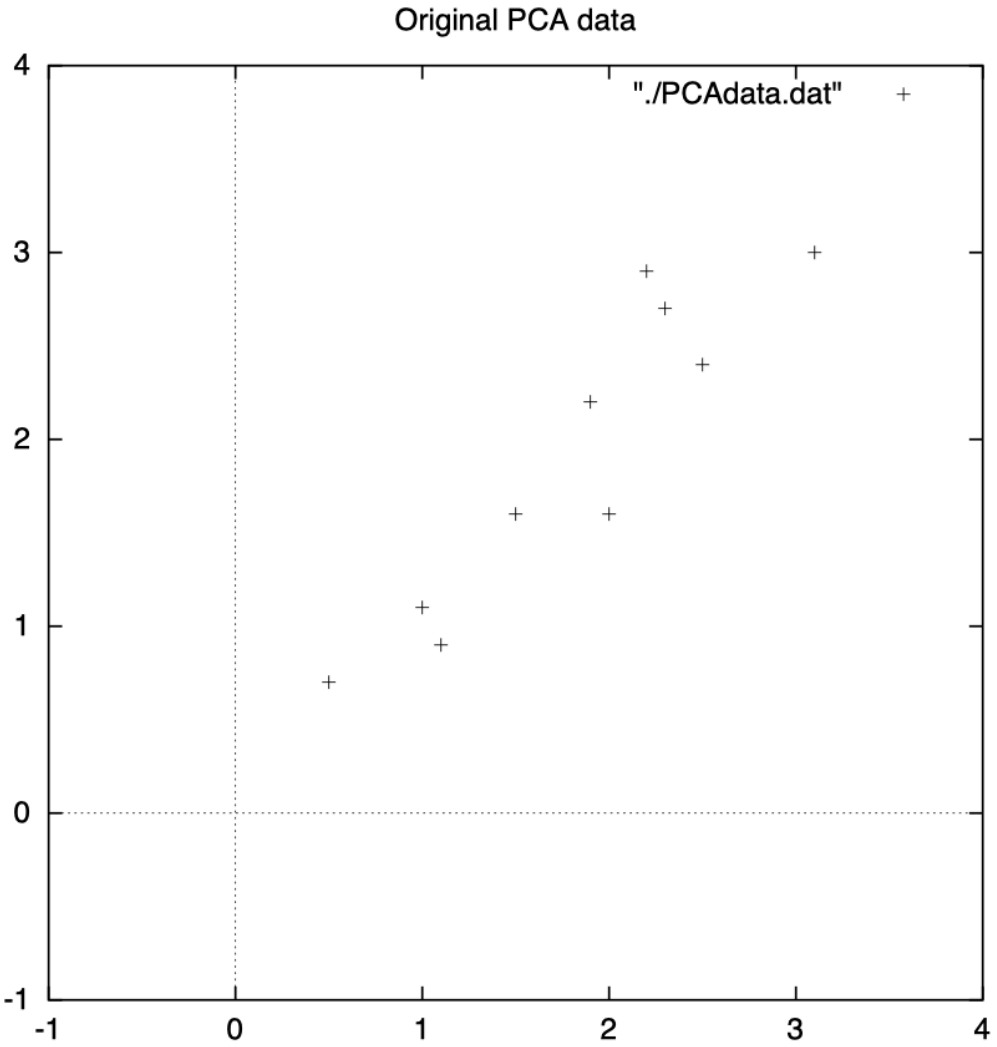
7. uncorrelated low-d data



Steps in PCA

1. Take the whole dataset consisting of *d+1 dimensions* and ignore the labels such that our new dataset becomes *d dimensional.*
2. Compute the *mean* for every dimension of the whole dataset. Shift the data to have 0 mean along each dimension.
3. Compute the *covariance matrix* of the whole dataset.
4. Compute *eigenvalues* and the corresponding *eigenvectors*.
5. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $d \times k$ *dimensional* matrix **W.**
6. Use this $d \times k$ *eigenvector matrix* to transform the samples onto the new subspace.

Step 1&2 Shift data to have 0 mean



Step 1&2 Shift data to have 0 mean

Data =

x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Step 3: Covariance Matrix

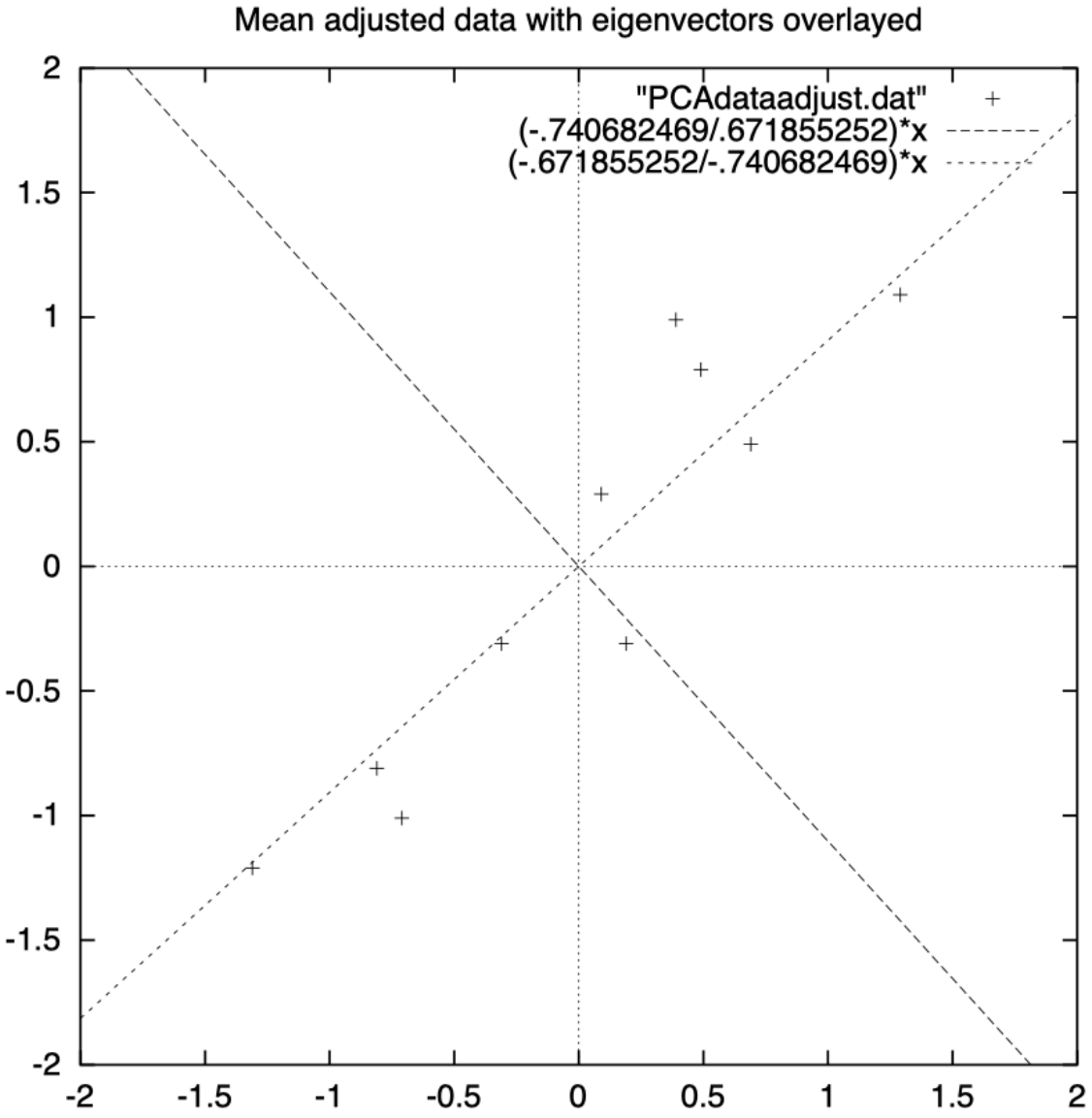
$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

Step 4: eigenvalues and eigenvectors

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

Step 4 result displayed on data



Step 5 Select the top K eigenvectors

- Sort all the eigenvectors in descending order of the corresponding eigenvalues:
  - If d=2 eigenvectors are used:

Column Feature Vector = $\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$

  - If d=1 eigenvector is used:

Column Feature Vector = $\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$

Step 6: Derive the new data

- Row feature vector = column feature vector^T

$$\begin{bmatrix} v_{11} & v_{21} & \cdots & v_{n1} \\ v_{12} & v_{22} & \cdots & v_{n2} \\ \vdots & \vdots & \cdots & \vdots \\ v_{1n} & v_{2n} & \cdots & v_{nn} \end{bmatrix}^T \rightarrow \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{bmatrix}$$

- Row data adjust = data adjust^T

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}^T \rightarrow \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{mn} \end{bmatrix}$$

n : size of dimensions, m: size of data

Step 6: Derive the new data

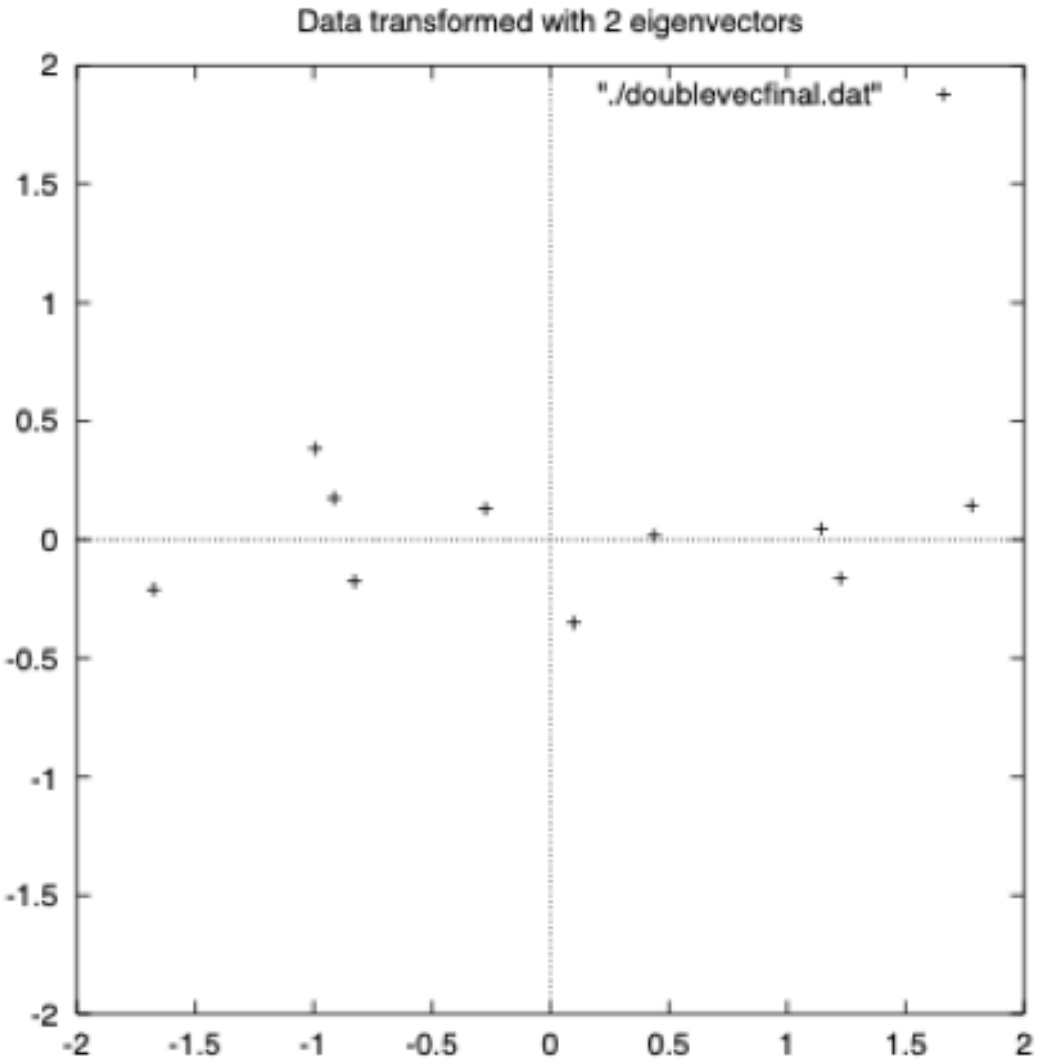
$$FinalData = RowFeatureVector \times RowDataAdjust$$

Final data = $\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{mn} \end{bmatrix}$

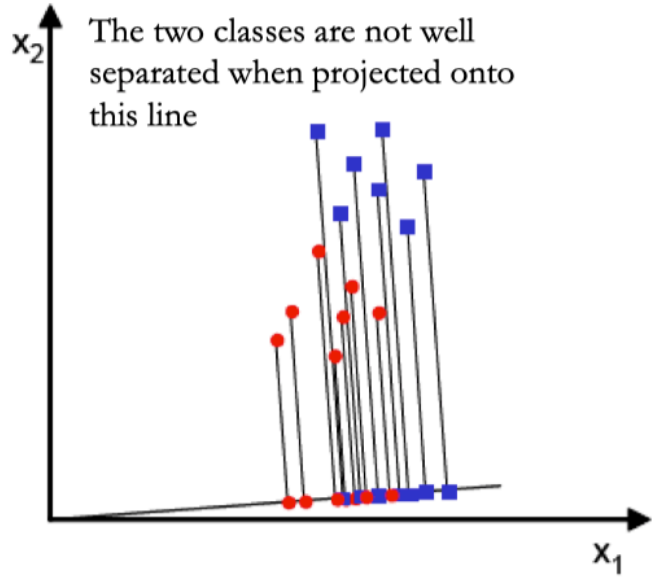
To reduce the dimension, pick the top k eigenvectors, where k < n

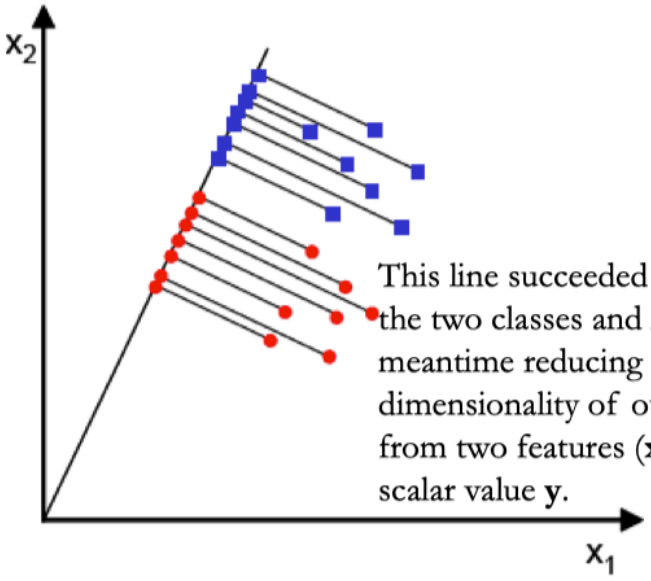
$$\begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{bmatrix} \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{mn} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{21} & \cdots & x_{m1} \\ x_{12} & x_{22} & \cdots & x_{m2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1k} & x_{2k} & \cdots & x_{mk} \end{bmatrix}$$

PCA result


Data transformed with 2 eigenvectors
"./doublevecfinal.dat"
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Linear Discriminant Analysis (LDA)





- Assume we have *m*-dimensional samples $\{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\}$, N_1 of which belong to $\boldsymbol{\omega}_1$ and N_2 belong to $\boldsymbol{\omega}_2$.
- We seek to obtain a scalar **y** by projecting the samples **x** onto a line (C-1 space, C = 2).

$$y = w^T x \quad \textit{where} \quad x = \begin{bmatrix} x_1 \\ . \\ . \\ x_m \end{bmatrix} \quad \textit{and} \quad w = \begin{bmatrix} w_1 \\ . \\ . \\ w_m \end{bmatrix}$$

  - where **w** is the projection vectors used to project **x** to **y**.
- **Of all the possible lines we would like to select the one that maximizes the separability of the scalars.**

LDA projected mean difference

- In order to find a good projection vector, we need to define a measure of separation between the projections.

- The mean vector of each class in **x** and **y** feature space is:

$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x \quad \text{and} \quad \tilde{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y = \frac{1}{N_i}\sum_{x\in\omega_i} w^T x$$

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

  - i.e. projecting **x** to **y** will lead to projecting the mean of **x** to the mean of **y**.

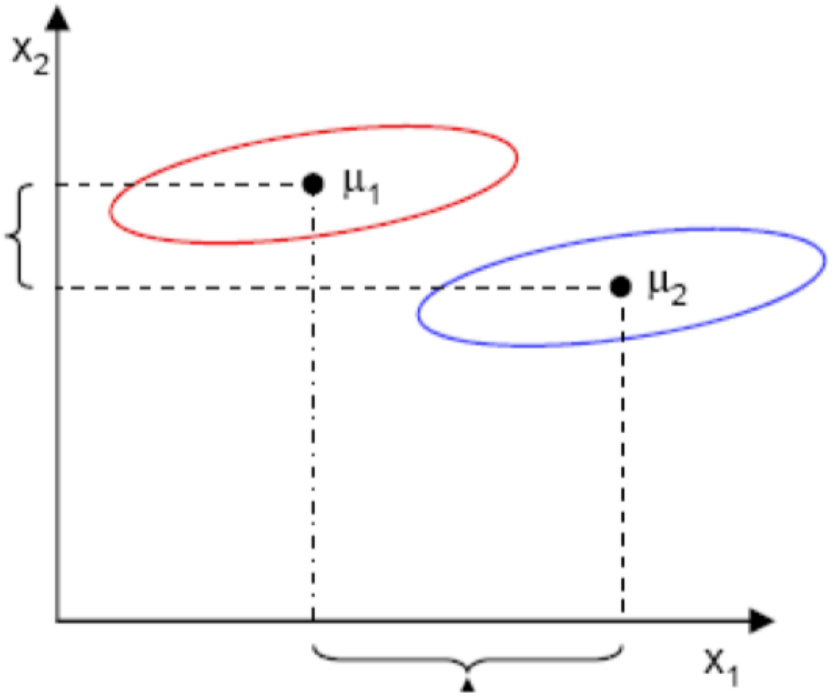
- We could then choose the distance between the projected means as our objective function

$$J(w) = \left|\tilde{\mu}_1 - \tilde{\mu}_2\right| = \left|w^T\mu_1 - w^T\mu_2\right| = \left|w^T\left(\mu_1 - \mu_2\right)\right|$$

LDA projected mean difference

- However, the distance between the projected means is not a very good measure since it does not take into account the standard deviation within the classes.

This axis yields better class separability



This axis has a larger distance between means

LDA scatter

- The solution proposed by Fisher is to maximize a function that represents the difference between the means, normalized by a measure of the within-class variability, or the so-called *scatter*.

- For each class we define the **scatter**, an equivalent of the variance, as; (sum of square differences between the projected samples and their class mean).

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2$$

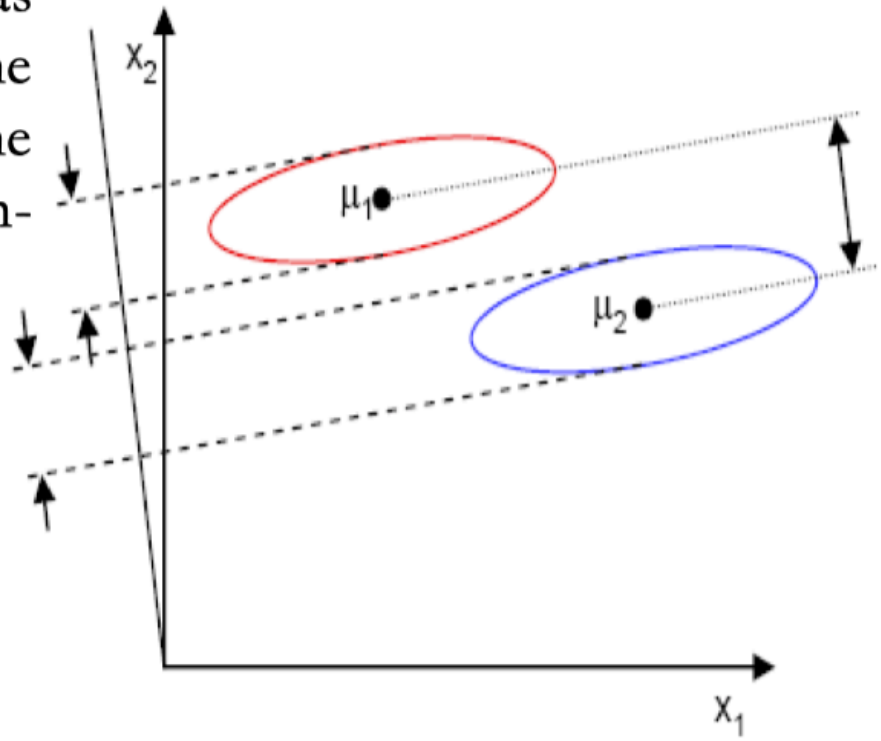
- $\tilde{s}_i^2$ measures the variability within class ω_i after projecting it on the y-space.

- Thus $\tilde{s}_1^2 + \tilde{s}_2^2$ measures the variability within the two classes at hand after projection, hence it is called *within-class scatter* of the projected samples.

LDA Objective function

- The Fisher linear discriminant is defined as the linear function $\mathbf{w}^T\mathbf{x}$ that maximizes the criterion function: (the distance between the projected means normalized by the within-class scatter of the projected samples.

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$



- Therefore, we will be looking for a projection where examples from the same class are projected very close to each other and, at the same time, the projected means are as farther apart as possible

LDA scatter matrices

- In order to find the optimum projection w^*, we need to express $J(w)$ as an explicit function of w.
- We will define a measure of the scatter in multivariate feature space **x** which are denoted as *scatter matrices*;

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

$$S_w = S_1 + S_2$$

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- Where $\mathbf{S_i}$ is the covariance matrix of class $\boldsymbol{\omega_i}$, and $\mathbf{S_w}$ is called the *within-class scatter matrix*.

Within class scatter matrix of project samples

- Now, the scatter of the projection **y** can then be expressed as a function of the scatter matrix in feature space **x**.

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in \omega_i} \left(w^T x - w^T \mu_i\right)^2$$

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

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

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^T S_1 w + w^T S_2 w = w^T (S_1 + S_2) w = w^T S_W w = \tilde{S}_W$$

Where $\tilde{S}_W$ is the within-class scatter matrix of the projected samples **y**.

Between class scatter of the projected samples

- Similarly, the difference between the projected means (in y-space) can be expressed in terms of the means in the original feature space (x-space).

$$\left(\tilde{\mu}_1 - \tilde{\mu}_2\right)^2 = \left(w^T\mu_1 - w^T\mu_2\right)^2$$

$$= w^T \underbrace{\left(\mu_1 - \mu_2\right)\left(\mu_1 - \mu_2\right)^T}_{S_B} w$$

$$= w^T S_B w = \tilde{S}_B$$

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

- The matrix $\mathbf{S_B}$ is called the *between-class scatter* of the original samples/feature vectors, while $\tilde{S}_B$ is the between-class scatter of the projected samples **y**.
- Since $\mathbf{S_B}$ is the outer product of two vectors, its rank is at most one.

LDA objective function

- We can finally express the Fisher criterion in terms of $\mathbf{S_W}$ and $\mathbf{S_B}$ as:

$$J(w) = \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

- Hence ***J(w)*** **is a measure of the difference between class means (encoded in the between-class scatter matrix) normalized by a measure of the within-class scatter matrix.**

Solve the objective function

- To find the maximum of $J(w)$, we differentiate and equate to zero.

$$\frac{d}{dw}J(w) = \frac{d}{dw}\left(\frac{w^T S_B w}{w^T S_W w}\right) = 0$$

$$\Rightarrow \left(w^T S_W w\right)\frac{d}{dw}\left(w^T S_B w\right) - \left(w^T S_B w\right)\frac{d}{dw}\left(w^T S_W w\right) = 0$$

$$\Rightarrow \left(w^T S_W w\right)2S_B w - \left(w^T S_B w\right)2S_W w = 0$$

Dividing by $2w^T S_W w$:

$$\Rightarrow \left(\frac{w^T S_W w}{w^T S_W w}\right)S_B w - \left(\frac{w^T S_B w}{w^T S_W w}\right)S_W w = 0$$

$$\Rightarrow S_B w - J(w)S_W w = 0$$

$$\Rightarrow S_W^{-1} S_B w - J(w)w = 0$$

Fisher's Linear Discriminant

- Solving the generalized eigen value problem

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

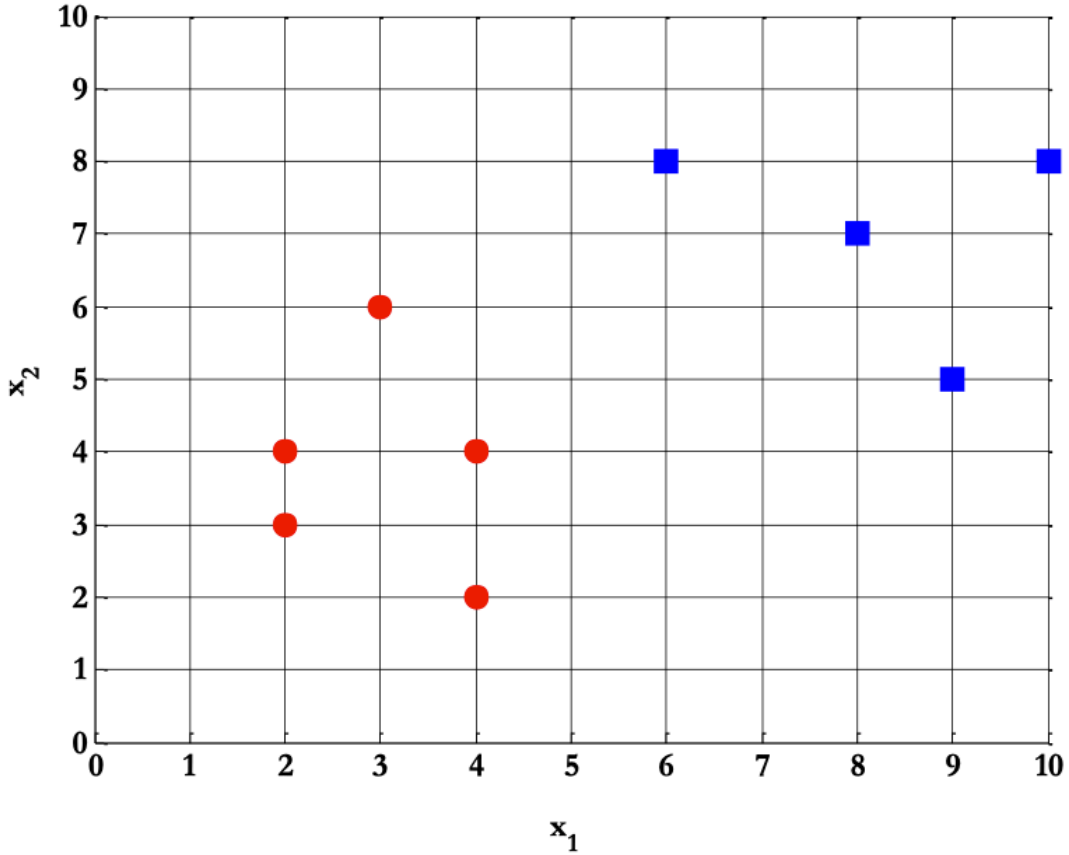
yields

$$w^* = \arg\max_w J(w) = \arg\max_w \left(\frac{w^T S_B w}{w^T S_W w} \right) = S_W^{-1} (\mu_1 - \mu_2)$$

- This is known as Fisher's Linear Discriminant, although it is not a discriminant but rather a specific choice of direction for the projection of the data down to one dimension.

An 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 : $\mathbf{X_2}=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$



An Example: Class mean

- The classes mean are :

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

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

An Example: Covariance matrix

- Covariance matrix of the first class:

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

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

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

An Example: Covariance matrix

- Covariance matrix of the second class:

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

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

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

An Example: Scatter matrix

- Within-class scatter matrix:

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

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

Compute Fisher Linear Discriminant

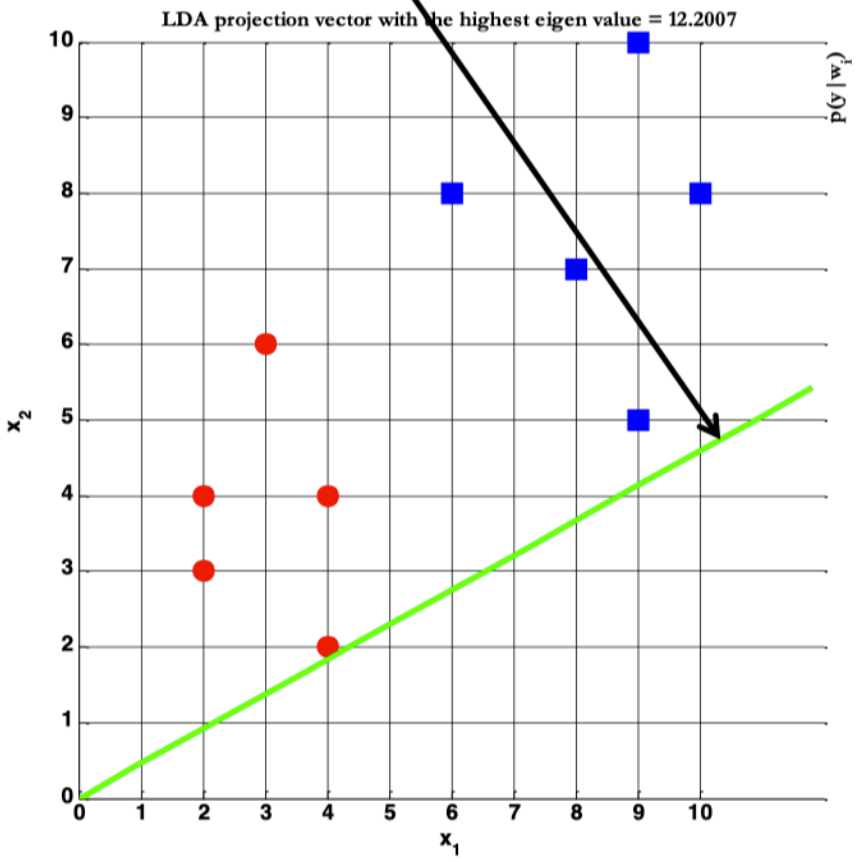
$$w^* = S_W^{-1}(\mu_1 - \mu_2) = \begin{pmatrix} 3.3 & -0.3 \\ -0.3 & 5.5 \end{pmatrix}^{-1} \left[\begin{pmatrix} 3 \\ 3.8 \end{pmatrix} - \begin{pmatrix} 8.4 \\ 7.6 \end{pmatrix} \right]$$

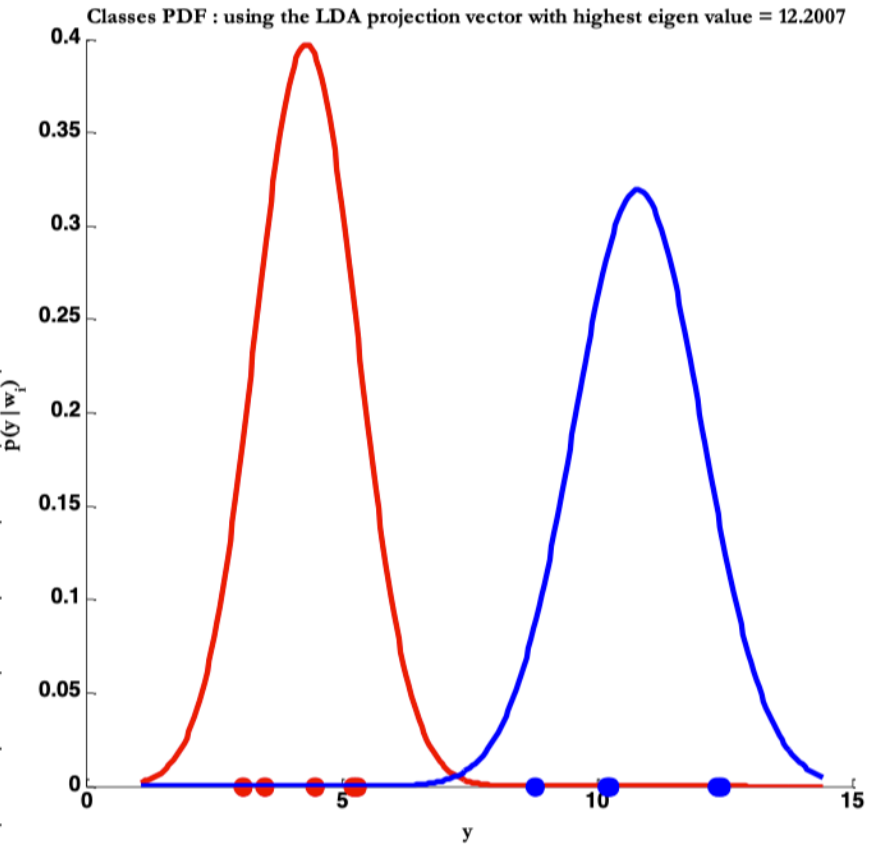
$$= \begin{pmatrix} 0.3045 & 0.0166 \\ 0.0166 & 0.1827 \end{pmatrix} \begin{pmatrix} -5.4 \\ -3.8 \end{pmatrix}$$

$$= \begin{pmatrix} 0.9088 \\ 0.4173 \end{pmatrix}$$

Project data with w

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





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

Solve the objective function using eigenvalue and eigenvectors

- Solving the generalized eigen value problem

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

- Using the same notation as PCA, **the solution will be the eigen vector(s) of** $S_X = S_W^{-1} S_B$

An Example: between class scatter matrix

- Between-class scatter matrix:

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

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

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

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

An Example: solved as a generalized eigenvalue problem

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

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

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

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

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

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

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

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

eigenvalues

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

and

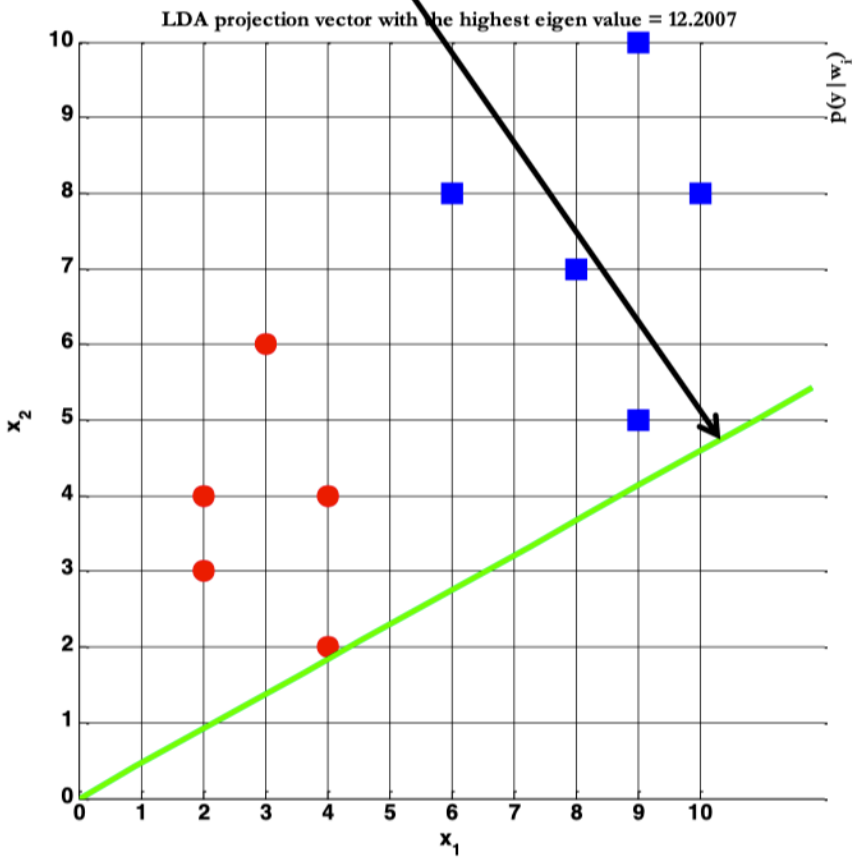
$$\begin{pmatrix} 9.2213 & 6.489 \\ 4.2339 & 2.9794 \end{pmatrix} w_2 = \underbrace{12.2007}_{\lambda_2} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$$

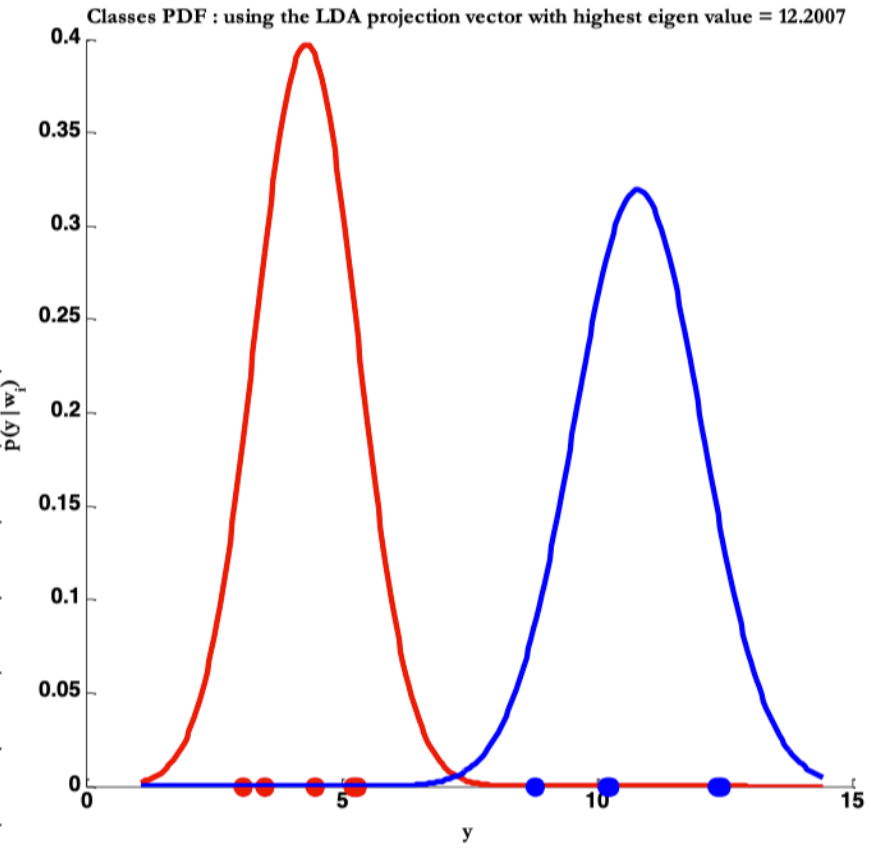
Thus;

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

Project data with w

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

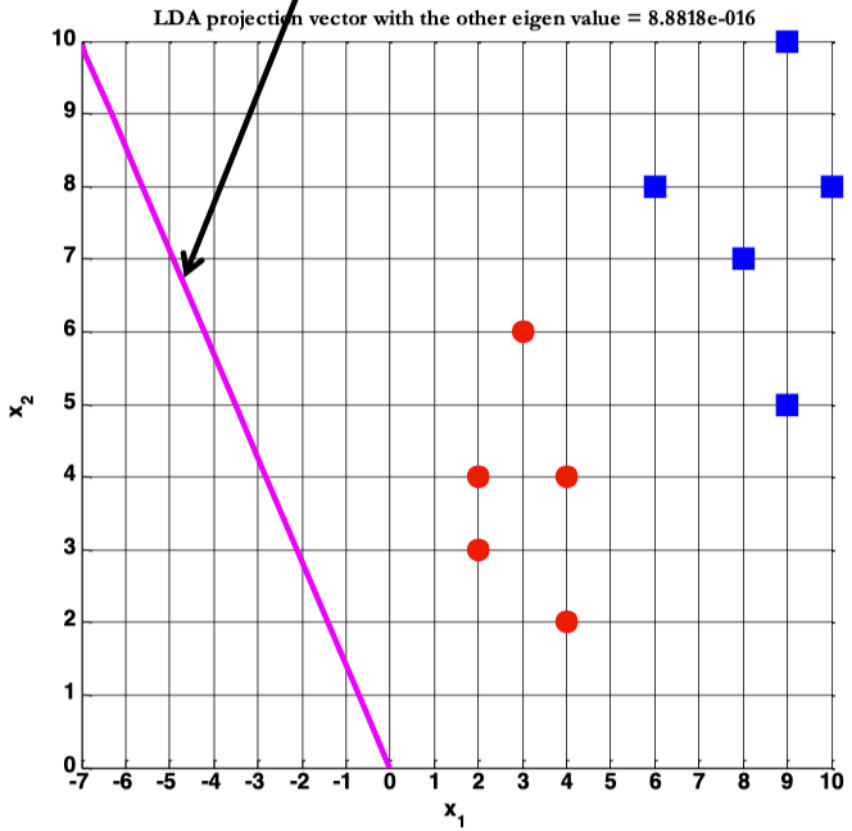


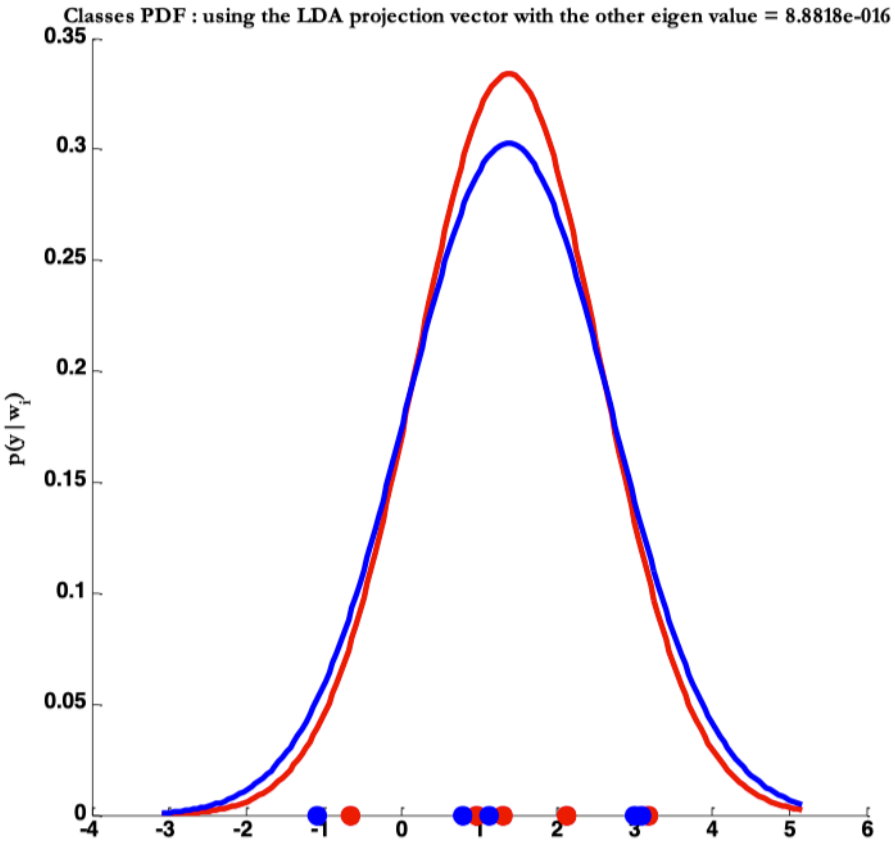


Using this vector leads to **good separability** between the two classes

Project data with w

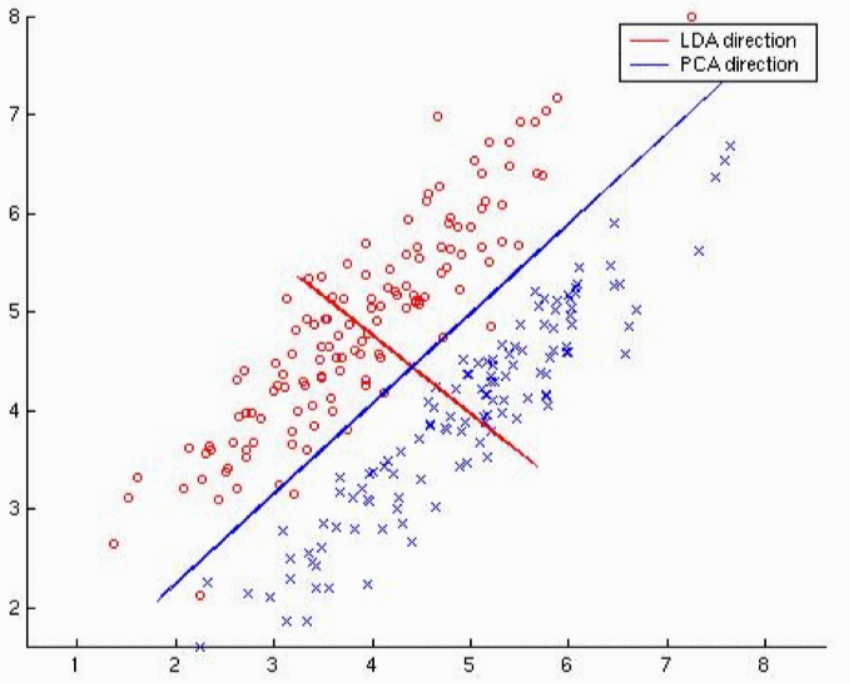
The projection vector corresponding to the **smallest** eigen value

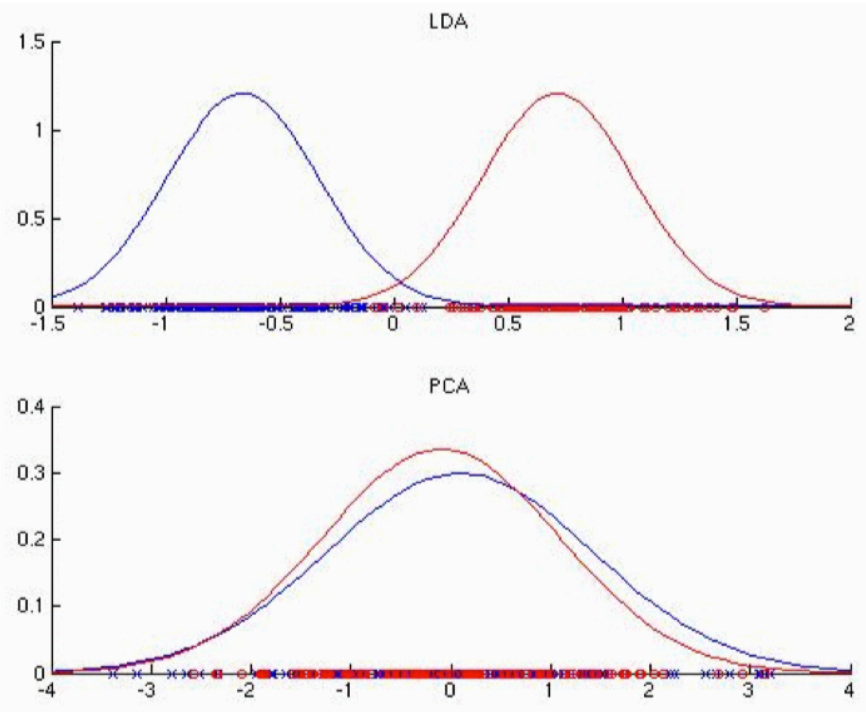




Using this vector leads to **bad separability** between the two classes

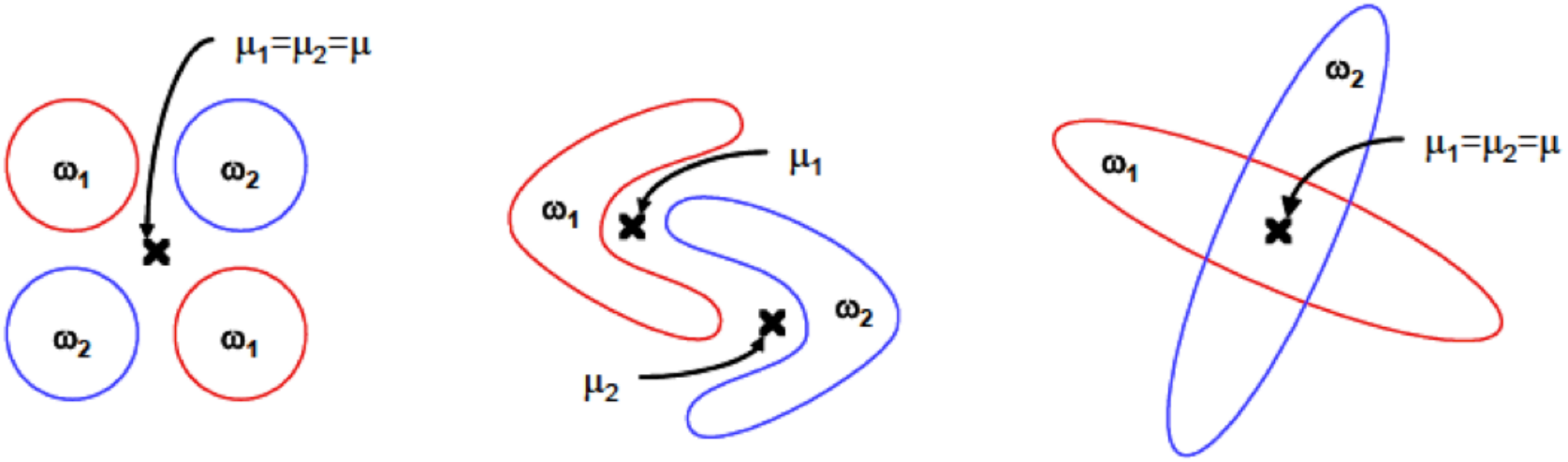
PCA vs LDA





Limitations of LDA

- **LDA is a parametric method since it assumes unimodal Gaussian likelihoods**
  - If the distributions are significantly non-Gaussian, the LDA projections will not be able to preserve any complex structure of the data, which may be needed for classification.



Limitations of LDA

- LDA will fail when the discriminatory information is not in the mean but rather in the variance of the data