Dimensionality Reduction

CSL603 - Fall 2017

Narayanan C Krishnan

ckn@iitrpr.ac.in

Outline

- Motivation
- Unsupervised Dimensionality Reduction
  - Principal Component Analysis
- Supervised Dimensionality Reduction
  - Linear Discriminant Analysis

Motivation (1)

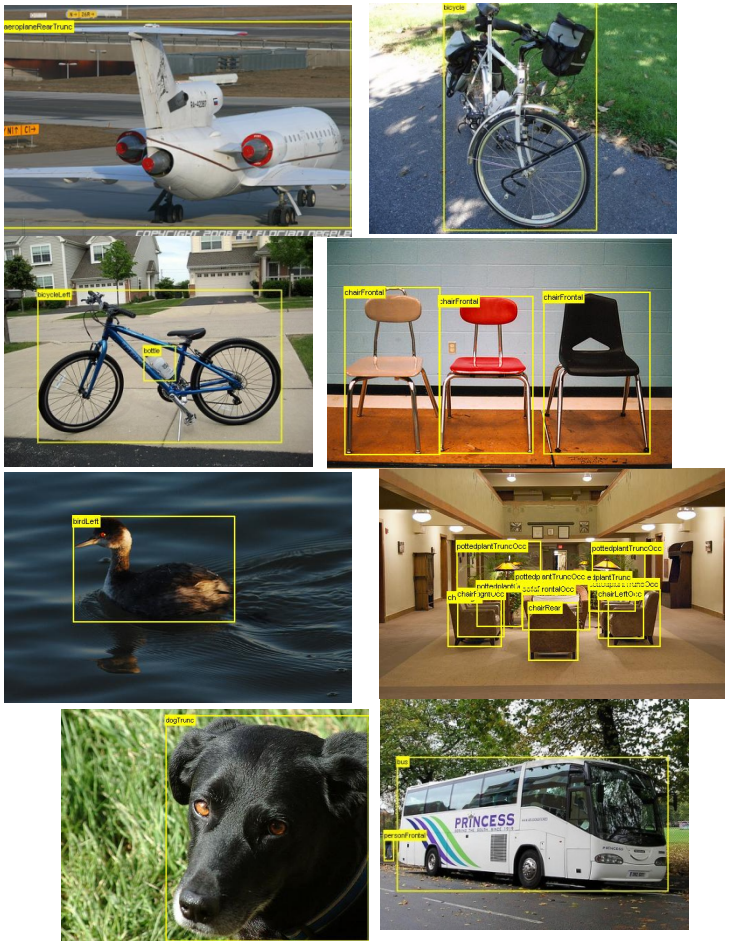
- Easy and convenient to collect data
  - Redundant and “not so useful” pieces of information
  - Simultaneously collecting for multiple objectives
  - Data accumulates at a rapid pace
    - Effective approach to downsize data
- Intrinsic dimension of the data may be small
  - Facial poses are governed by two-three degrees of freedom
- Visualization
  - Projecting high-dimensional data onto 2-3 dimensions

Motivation (2)

- Curse of Dimensionality
  - Ineffectiveness of most machine learning and data mining techniques on high-dimensional data
- Data Compression
  - Efficient storage and retrieval
- Noise removal

Examples of High Dimensional Data (1)

- Computer Vision



Examples of High Dimensional Data (2)

- Document/Text Analysis

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Documents	Terms: T_1	T_2	.....	T_N
D_1	12	0	.....	6
D_2	3	10	.....	28
⋮	⋮			⋮
D_M	0	11	.....	16

- Classify the documents
  - Thousands of documents
  - Thousands of terms

20 Newsgroups data
- 20K documents
- 40K features!

Dimensionality Reduction Approaches

- Feature extraction(reduction)
  - Subspace learning: mapping high dimensional data onto a lower dimensional space
    - Linear
    - Non-Linear
- Feature selection
  - Selects an optimal subset of features from the existing set according to an objective function

Representative Algorithms

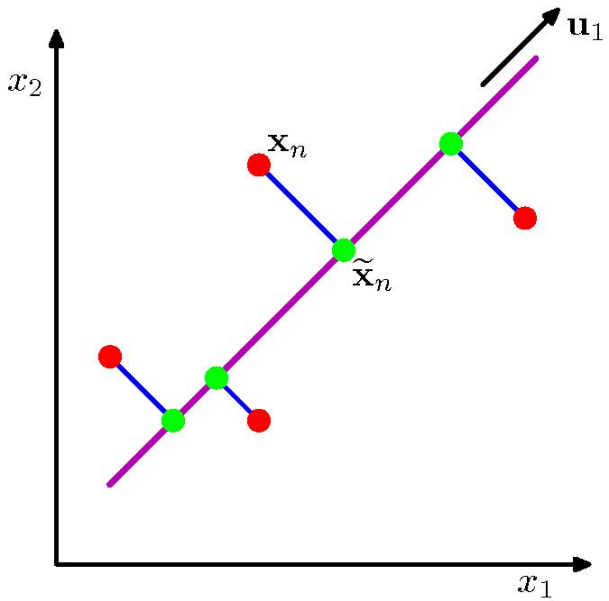
- Unsupervised
  - Principal Component Analysis (PCA)
    - Kernel PCA, Probabilistic PCA
  - Independent Component Analysis (ICA)
  - Latent Semantic Indexing (LSI)
  - Manifold Learning
- Supervised
  - Linear Discriminant Analysis
  - Canonical Correlation Analysis
  - Partial Least Squares

Principal Component Analysis (PCA)

- Basic Idea
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information
  - The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.
- Also known as *Karhunen-Loève* transform
- A very popular feature extraction technique

PCA - Formulation

- Objective:
  - Minimize error $\sum_{i=1}^{N} \|x_i - \tilde{x}_i\|_2$ (minimum projection error)
  - Maximum variance of the projected data
- Theoretically both are equivalent



PCA – Maximum Variance Formulation (1)

- Consider set of data points $\{x_i\}$ where $i = 1, \ldots, N$ and $x_i \in \mathbb{R}^D$
  - Mean of the original data: $\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$
- Goal: Project data onto $\mathrm{K} < D$ dimensional space while maximizing the variance of the projected data
- To begin with, consider $\mathrm{K} = 1$:
  - Let u_1 be the direction of the projection.
  - Set $\|u_1\| = 1$, as it is only the direction that is important
  - Projected data: $u_1^T x_i$ and Projected mean: $u_1^T \bar{x}$

PCA – Maximum Variance Formulation (2)

- Covariance of original data:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

PCA – Maximum Variance Formulation (3)

- Covariance of original data:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

- Variance of the projected data:

PCA – Maximum Variance Formulation (4)

- Covariance of original data:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

- Variance of the projected data:

$$\frac{1}{N}\sum_{i=1}^{N}\{u_1^T x_i - u_1^T \bar{x}\}^2 = \frac{1}{N}\sum_{i=1}^{N}\{u_1^T (x_i - \bar{x})\}^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} u_1^T (x_i - \bar{x})(x_i - \bar{x})^T u_1 = u_1^T \Sigma\, u_1$$

- Goal: maximizing variance of the projected data

$$\max_{u_1} u_1^T \Sigma\, u_1 \text{ such that } \|u_1\|_2 = 1$$

PCA – Maximum Variance Formulation (5)

- Using Lagrange multipliers

$$\max_{u_1} \quad u_1^T \Sigma \, u_1 + \lambda_1 (1 - u_1^T u_1)$$

- By setting the derivative wrt u_1 equal to 0

PCA – Maximum Variance Formulation (6)

- Using Lagrange multipliers

$$\max_{u_1} \; u_1^T \Sigma \, u_1 + \lambda_1 (1 - u_1^T u_1)$$

- By setting the derivative wrt u_1 equal to 0

$$\Sigma u_1 = \lambda_1 u_1$$

- u_1 must be an ***eigenvector*** of Σ ! And $u_1^T \Sigma \, u_1 = \lambda_1$ implying variance is maximized by choosing the eigenvector associated with the largest eigenvalue.
- u_1 corresponds to the first principal component.

PCA – Maximum Variance Formulation (7)

- Second principal direction: Maximize variance $u_2^T \Sigma u_2$, subject to $\|u_2\|_2 = 1$ and $u_2^T u_1 = 0$

PCA – Maximum Variance Formulation (8)

- Second principal direction: Maximize variance $u_2^T \Sigma u_2$, subject to $\|u_2\|_2 = 1$ and $u_2^T u_1 = 0$

$$\text{Max } u_2^T \Sigma u_2 + \lambda_2(1 - u_2^T u_2) + \eta(u_2^T u_1)$$

- Setting the derivative wrt u_2 equal to 0

PCA – Maximum Variance Formulation (9)

- Second principal direction: Maximize variance $u_2^T \Sigma u_2$, subject to $\|u_2\|_2 = 1$ and $u_2^T u_1 = 0$

$$\text{Max } u_2^T \Sigma u_2 + \lambda_2(1 - u_2^T u_2) + \eta(u_2^T u_1)$$

- Setting the derivative wrt u_2 equal to 0

$$2\Sigma u_2 - 2\lambda_2 u_2 + \eta u_1 = 0$$

- $\Sigma u_2 = \lambda_2 u_2$ - choose u_2 as the eigenvector of Σ with the second largest eigenvalue λ_2
- And so on …

u_3 $\quad u_3^T \Sigma u_3 \quad \|u_3\|_2 = 1, \; u_3^T u_2 = 0 \quad u_3^T u_1 = 0$

PCA – Maximum Variance Formulation (10)

- Final Solution:
  - Eigenvalue decomposition (SVD) of the covariance matrix

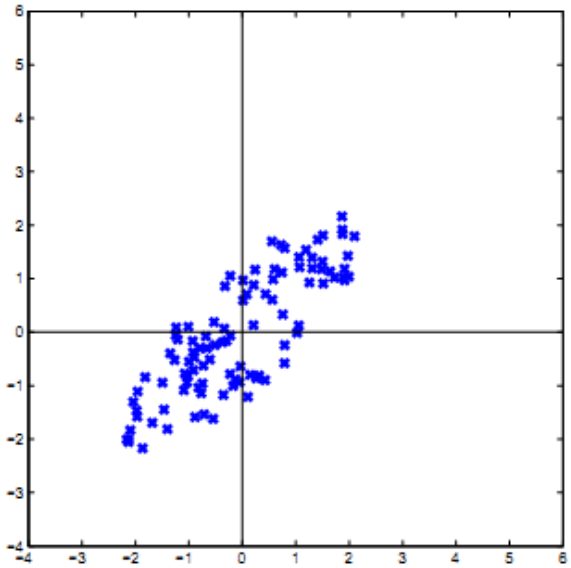
$$\Sigma = U\Lambda U^T$$

  - For $\mathrm{K} < D$ dimensional projection space:
  - Choose K eigenvectors $\{u_1, u_2, \ldots, u_K\}$with the largest associated eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$

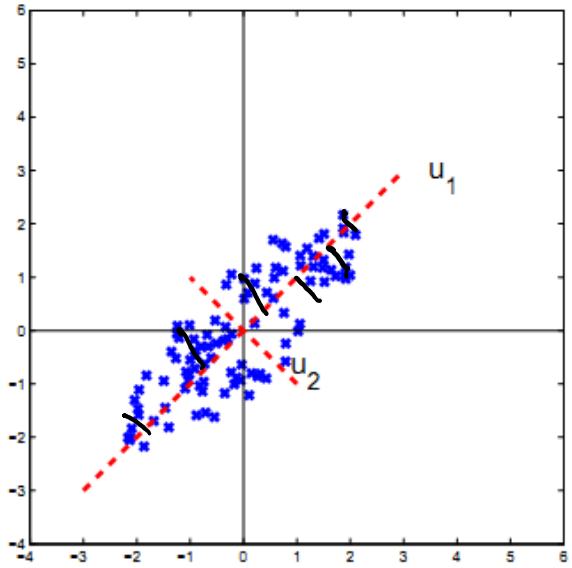
PCA - Algorithm

1. Create ~~$D \times N$~~ data matrix X, with one row vector x_i per data point
2. Subtract mean $\bar{x}$ from each ~~column~~ vector x_i in X
3. $\Sigma \leftarrow$ covariance matrix of X
4. Find eigenvectors U and eigenvalues Λ of Σ
5. PC's U_K the K eigenvectors with largest eigenvalues
6. Transformed data $Z = U^T X$

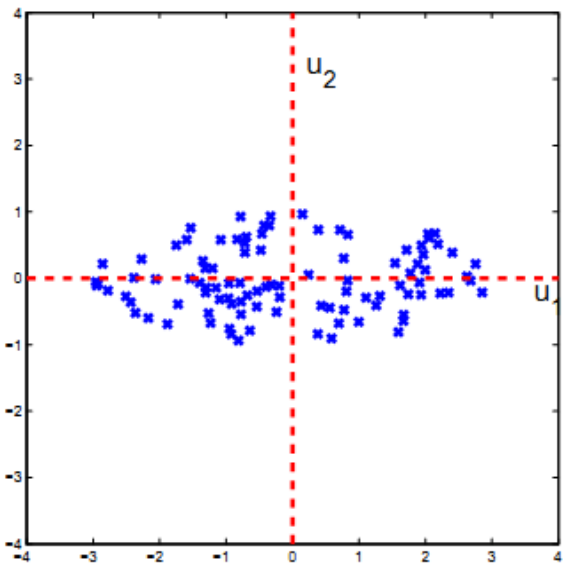
PCA – Illustration (1)



Original data

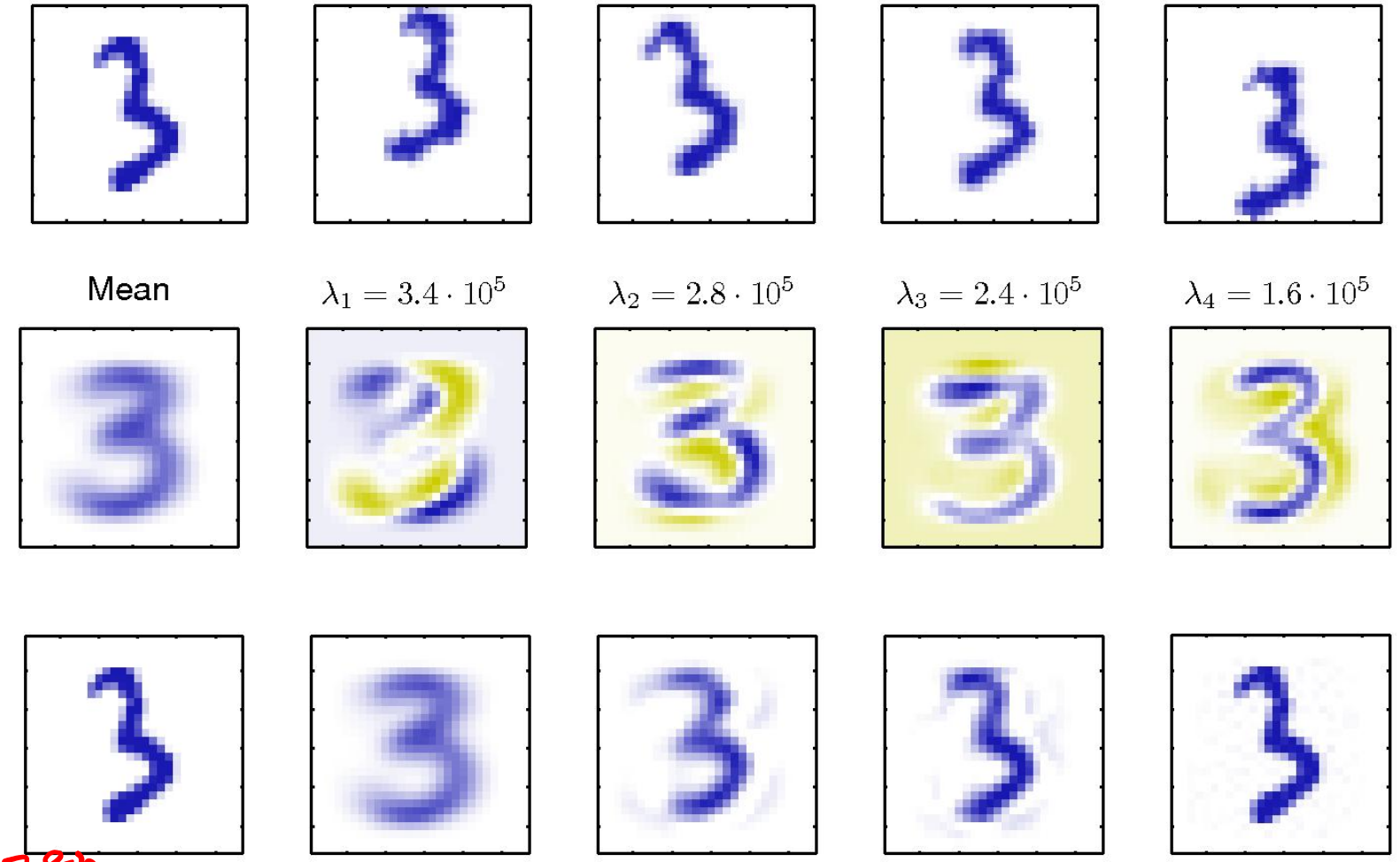


Principal components (PC)



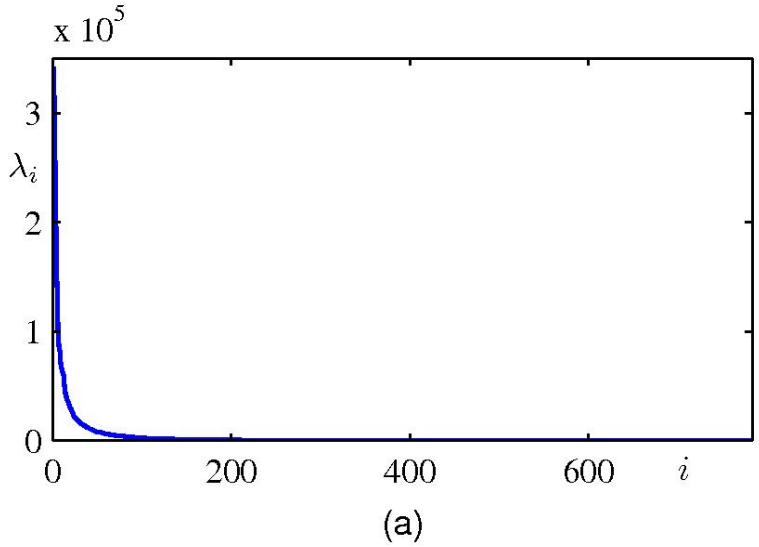
Original data projected into the PC space

PCA – Illustration (2)

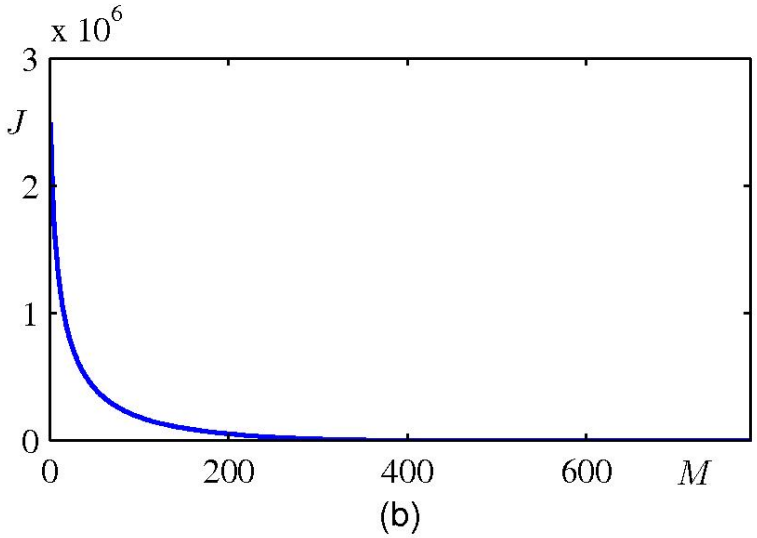


PCA – Model Selection

- How many principal components to select?
- Heuristic: Detect the knee in the eigenvalue spectrum and retain the large eigenvalues



Eigenvalue spectrum for off-line digits dataset



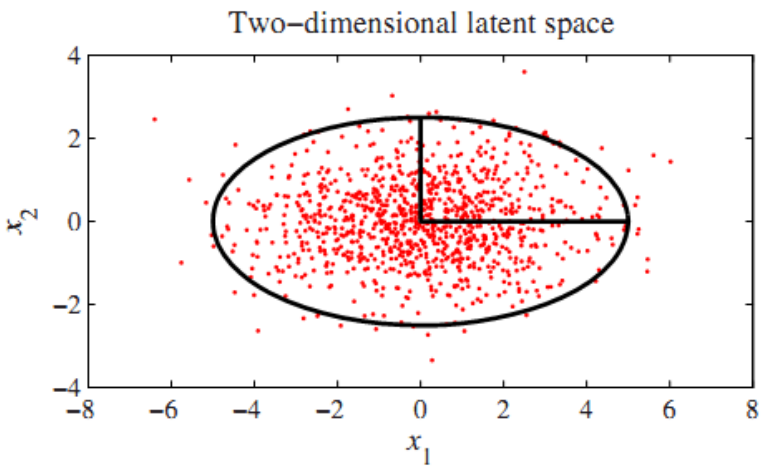
Reconstruction error by projecting the data into M PC's

Limitations of PCA (1)

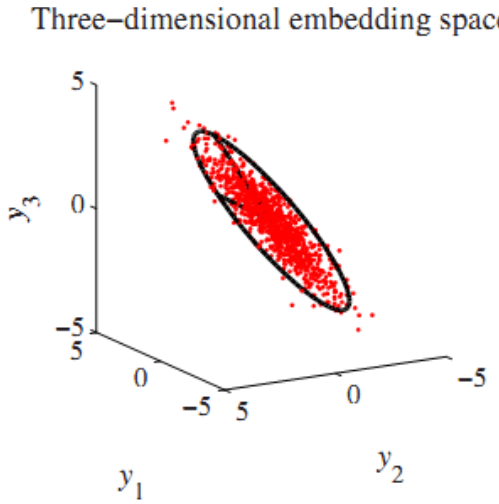
- Non- parametric
  - No probabilistic model for observed data
- The co-variance matrix needs to be calculated
  - Can be computation intensive for datasets with a high number of dimensions
- Does not deal with missing data
  - Incomplete data must either be discarded or imputed using ad-hoc methods
- Outliers in the data can unduly affect the analysis
- Batch mode algorithm

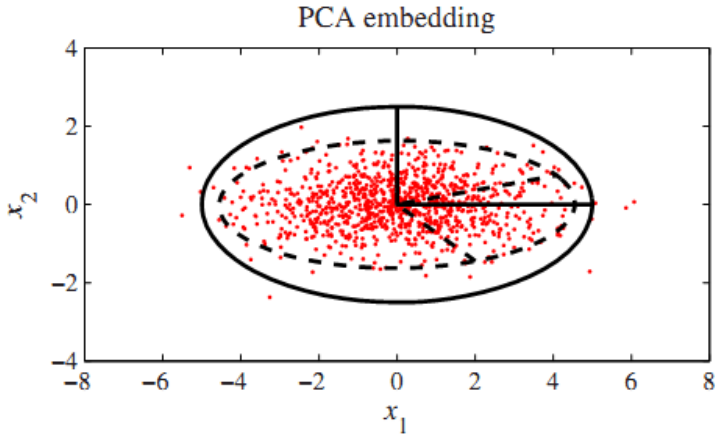
Assumptions of PCA

- Gaussian Assumption with linear embedding



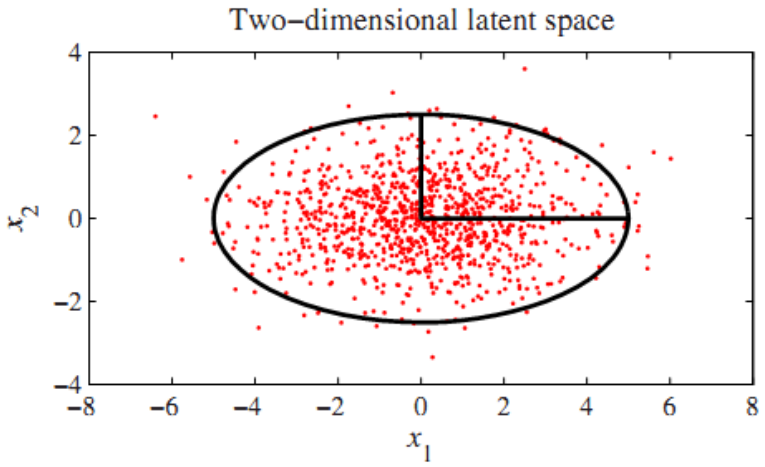
$$\mathbf{W} = \begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.5 \\ 0.7 & 0.3 \end{bmatrix}$$



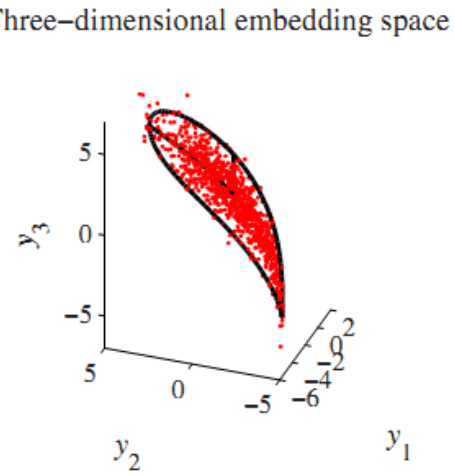


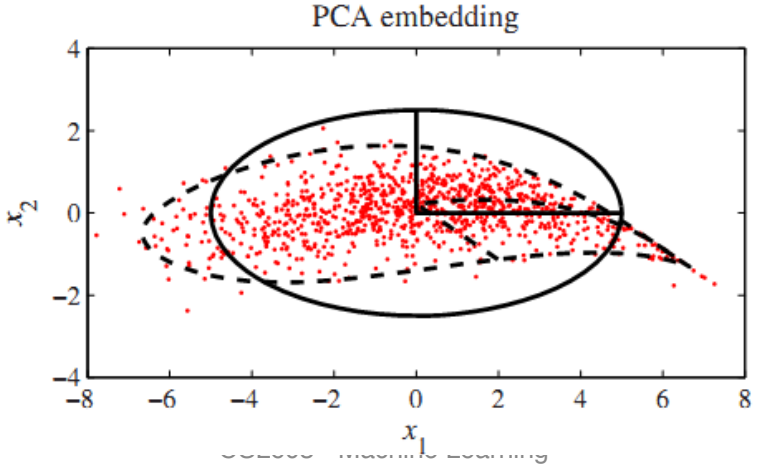
Limitations of PCA (2)

- Non-linear embedding



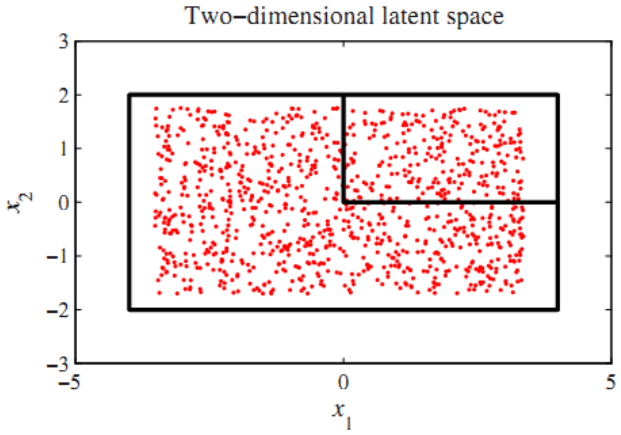
$$\mathbf{y} = \begin{bmatrix} 4\cos(\frac{1}{4}x_1) \\ 4\sin(\frac{1}{4}x_1) \\ x_1 + x_2 \end{bmatrix}$$



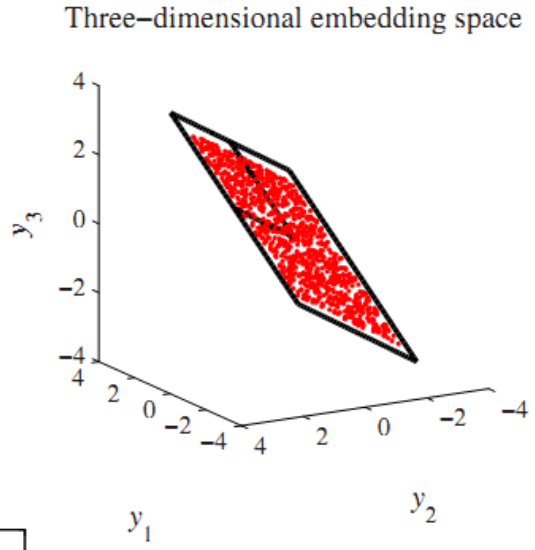


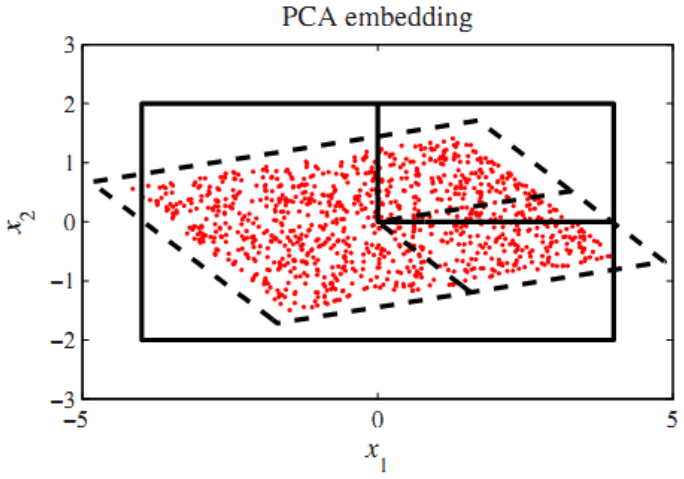
Limitations of PCA (3)

- Non-Gaussian Assumption



$$\mathbf{W} = \begin{bmatrix} 0.2 & 0.8 \\ 0.4 & 0.5 \\ 0.7 & 0.3 \end{bmatrix}$$





Nonlinear PCA using Kernels

- Traditional PCA applies linear transformation
  - May not be effective for nonlinear data
- Solution: apply nonlinear transformation to the high dimensional space

$$\phi: x \rightarrow \phi(x)$$

- Computational efficiency through kernel trick

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

- Compute the eigenvectors of K rather than the sample covariance or covariance in the high-dimensional space

$\{x_i\}_{i=1}^{N} \quad \bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i = 0$

$\Sigma = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T \quad \lambda u = \Sigma u = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T u$

$= \frac{1}{N}\sum_{i=1}^{N} (x_i^T u)\, x_i$

u : e.v. $\rightarrow \lambda$

$\therefore \quad u = \sum_{i=1}^{N} \left(\frac{x_i^T u}{\lambda N}\right) x_i$

let $\frac{x_i^T u}{\lambda N} = \alpha_i$ (a scalar)

then $\boxed{u = \sum_{i=1}^{N} \alpha_i x_i}$

$\Sigma u = \lambda u \qquad x_i^T \Sigma u = \lambda x_i^T u$

$= x_i^T \frac{1}{N}\sum_{j=1}^{N} x_j x_j^T \; u = \lambda x_i^T u$

$\Rightarrow x_i^T \frac{1}{N}\sum_{j=1}^{N} x_j x_j^T \sum_{k=1}^{N} \alpha_k x_k = \lambda x_i^T \sum_{k=1}^{N} \alpha_k x_k$

$= \frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N} \alpha_k x_i^T x_j x_j^T x_k = \lambda \sum_{k=1}^{N} \alpha_k x_i^T x_k$

$$(x_i x_i^T u) = \begin{bmatrix} x_{i1}x_{i1} & x_{i1}x_{i2} & \cdots & x_{i1}x_{iD} \\ x_{i2}x_{i1} & x_{i2}x_{i2} & \cdots & x_{i2}x_{iD} \\ \vdots & & & \\ x_{iD}x_{i1} & x_{iD}x_{i2} & \cdots & x_{iD}x_{iD} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_D \end{bmatrix}$$

$$= \begin{bmatrix} x_{i1}x_{i1}u_1 + x_{i1}x_{i2}u_2 + \cdots + x_{i1}x_{iD}u_D \\ x_{i2}x_{i1}u_1 + x_{i2}x_{i2}u_2 + \cdots + x_{i2}x_{iD}u_D \\ \vdots \\ x_{iD}x_{i1}u_1 + x_{iD}x_{i2}u_2 + \cdots + x_{iD}x_{iD}u_D \end{bmatrix}$$

$$= \begin{bmatrix} (x_{i1}u_1 + x_{i2}u_2 + \cdots + x_{iD}u_D)\, x_{i1} \\ (x_{i2}u_1 + x_{i2}u_2 + \cdots + x_{iD}u_D)\, x_{i2} \\ \vdots \\ (x_{i1}u_1 + x_{i2}u_2 + \cdots + x_{iD}u_D)\, x_{iD} \end{bmatrix} = (x_i^T u) \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix}$$

$$= (x_i^T u)\, x_i$$

$$\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\alpha_k x_i^T x_j x_j^T x_k = \lambda\sum_{k=1}^{N}\alpha_k x_i^T x_k$$

considering for all the N points, we get:

$$\frac{1}{N}K^2\alpha = \lambda\alpha K$$

α - N-dimensional vector

K - kernel matrix

$K(i,j) = x_i^T x_j$

$$\Rightarrow K\alpha = N\lambda\alpha \quad \text{—(1)}$$

reduces to eigenvalue decomposition of K.

The unit norm constraint on u $\Rightarrow \|u\|_2^2 = 1 \Rightarrow u^T u = 1$

$$\Rightarrow \sum_{i=1}^{N}\alpha_i x_i^T \sum_{j=1}^{N}\alpha_j x_j = 1 \Rightarrow \sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j x_i^T x_j = 1$$

$$\Rightarrow \alpha K\alpha = 1 \text{ - using (1)}$$

$$\alpha N\lambda\alpha = 1 \quad \alpha^2 = \frac{1}{N\cdot\lambda}.$$

For a test data point x, the projection is

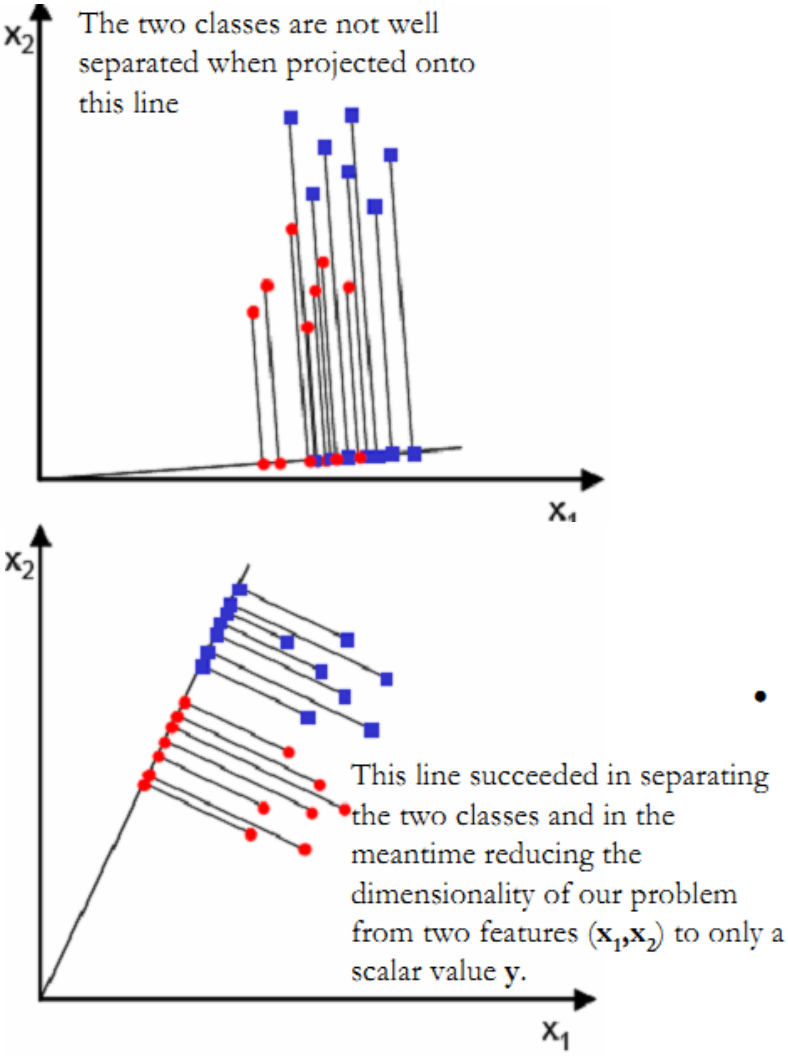
$u^T x = \sum_{i=1}^{N}\alpha_i x_i^T x$, here too we can use the kernel fn. K.

This derivation assumes that the data is centered. However centering the data in the Z space is tricky without knowing ϕ. This is achieved through some operations on K. (Refer reading material).

Linear Discriminant Analysis (LDA)

- Objective: perform dimensionality reduction, while preserving as much of the class discriminatory information as possible
  - Supervised dimensionality reduction
- Seeks to find directions along which the classes are best separated

LDA - Illustration



- Consider set of data points $\{x_i\}$ where $i = 1, \ldots, N$ and $x_i \in \mathbb{R}^D$, N_1 of which belong to C_1 and N_2 belong to C_2.
- Obtain scalar z by projecting samples x onto a line ($C - 1$ dimensional space)

$$z = w^T x$$

- Find w that maximizes the separability of the scalars

LDA Formulation – Two Classes (1)

- 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 the original and transformed space is

$$\mu_k = \frac{1}{N_k} \sum_{x_i \in C_k} x_i$$

$$\bar{\mu}_k = \frac{1}{N_k} \sum_{z_i \in C_k} z_i$$

LDA Formulation – Two Classes (2)

- 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 the original and transformed space is

$$\mu_k = \frac{1}{N_k}\sum_{x_i \in C_k} x_i,\ \bar{\mu}_k = \frac{1}{N_k}\sum_{z_i \in C_k} z_i = \frac{1}{N_k}\sum_{x_i \in C_k} w^T x_i$$

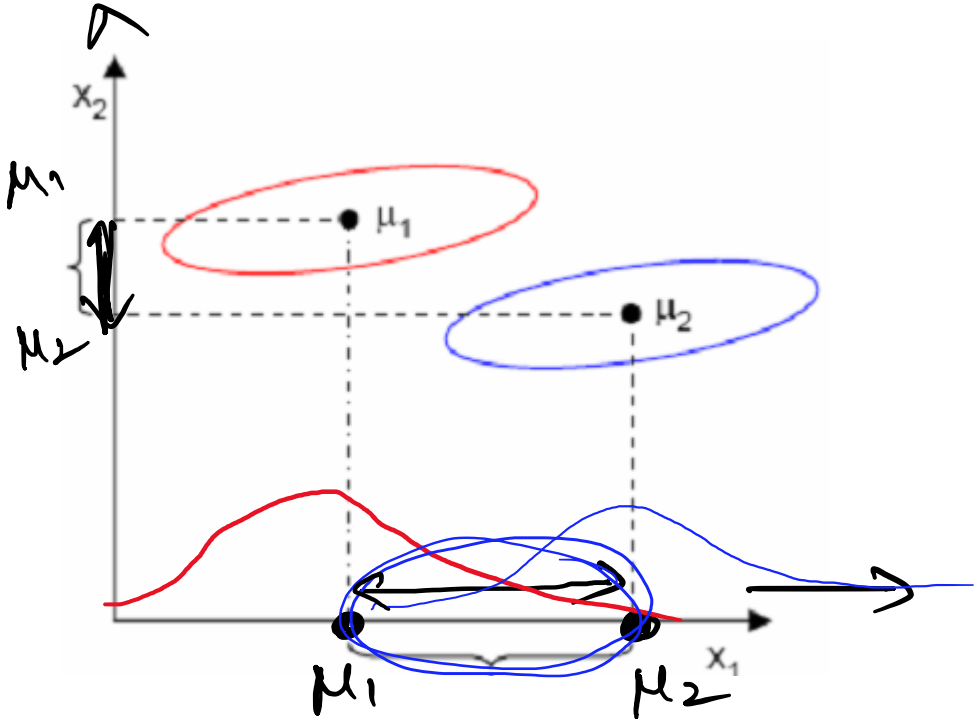
- We could then choose the distance between the projected means as our objective function

$$J(w) = \|\bar{\mu}_1 - \bar{\mu}_2\|^2 = \|w^T(\mu_1 - \mu_2)\|^2$$

LDA Formulation – Two Classes (3)

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

- Is this a good measure?



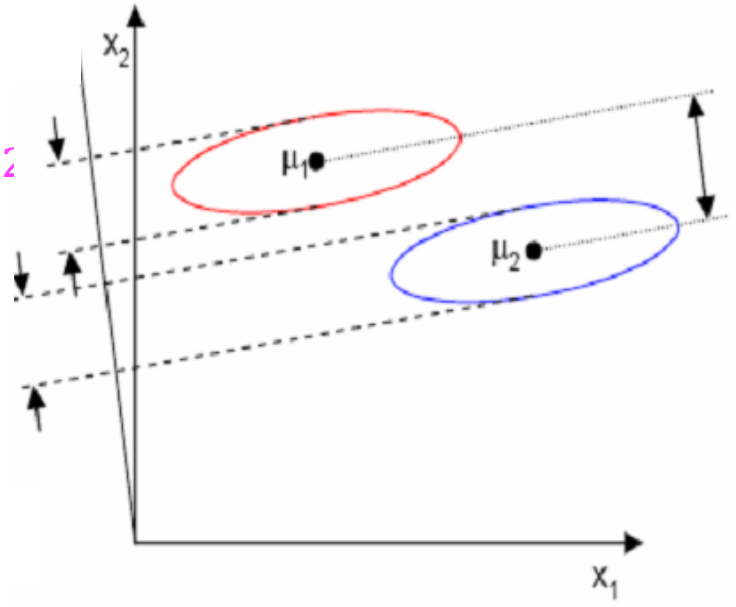
LDA Formulation – Two Classes (4)

- Proposed by Fisher (Fisher Discriminant Analysis)
  - Maximize a function that represents the difference between the means, normalized by a measure of the within-class variability (scatter)

$$J(w) = \frac{\|\bar{\mu}_1 - \bar{\mu}_2\|^2}{\bar{S}_1 + \bar{S}_2}$$

  - Where

$$\bar{S}_k = \sum_{z_i \in C_k} (z_i - \bar{\mu}_k)^2$$



LDA Formulation – Two Classes (5)

- 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 the multivariate feature space x, denoted through *scatter matrices*

$$S_k = \sum_{x \in k} (x - \mu_k)(x - \mu_k)^T$$

$$S_w = S_1 + S_2$$

  - S_w is called the *within-class scatter matrix*

LDA Formulation – Two Classes (6)

- Scatter of the transformed data z can then be expressed as a function of the scatter matrix of the original data x.

$$\bar{S}_k = \sum_{z_i \in C_k} (z_i - \bar{\mu}_k)^2$$

LDA Formulation – Two Classes (7)

- Scatter of the transformed data z can then be expressed as a function of the scatter matrix of the original data x.

$$\bar{S}_k = \sum_{z_i \in C_k} (z_i - \bar{\mu}_k)^2 = w^T S_k w$$

$$\bar{S}_1 + \bar{S}_2 =$$

LDA Formulation – Two Classes (8)

- Scatter of the transformed data z can then be expressed as a function of the scatter matrix of the original data x.

$$\bar{s}_k^2 = \sum_{z_i \in C_k} (z_i - \bar{\mu}_i)^2 = w^T S_k w$$

$$\bar{S}_1 + \bar{S}_2 = w^T S_1 w + w^T S_2 W = w^T S_w w$$

- Difference in the projected means in the transformed space can be expressed in terms of the means in the original feature space

$$(\bar{\mu}_1 - \bar{\mu}_2)^2 = (w^T \mu_1 - w^T \mu_2)^2$$
$$= w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = w^T S_B w$$

LDA Formulation – Two Classes (9)

- Express Fisher criteria in terms of S_W and S_B as:

$$J(w) = \frac{\|\bar{\mu}_1 - \bar{\mu}_2\|^2}{\bar{S}_1 + \bar{S}_2} = \frac{w^T S_B w}{w^T S_w w}$$

- To find the maximum of $J(w)$, differentiate it wrt w and equate to 0

$$w^T S_w w \cdot 2 S_B w - w^T S_B w \cdot 2 S_w w = 0$$

$$\Rightarrow \frac{w^T S_w w}{w^T S_w w} \cdot 2 S_B w - \frac{w^T S_B w}{w^T S_w w} \cdot 2 S_w w = 0 \qquad J(w)$$

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

$$S_B w = \lambda S_w w \quad (\lambda = J(w))$$

generalized eigen value problem.

LDA Formulation – Two Classes (10)

- Express Fisher criteria in terms of S_W and S_B as:

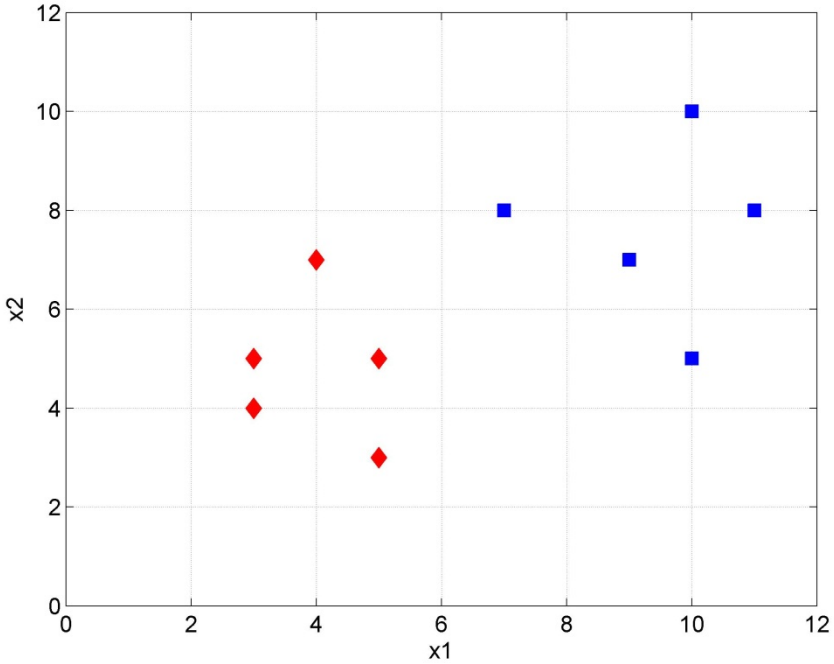
$$J(w) = \frac{\|\bar{\mu}_1 - \bar{\mu}_2\|^2}{\overline{S}_1 + \overline{S}_2} = \frac{w^T S_B w}{w^T S_w w}$$

- To find the maximum of $J(w)$, differentiate it wrt w and equate to 0

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

- If $\lambda = J(w)$, then $S_W^{-1} S_B w = \lambda w$
- w is in the direction of $S_W^{-1}(\mu_1 - \mu_2)$- *Fisher's Linear Discriminant*.
- Using the same notation as PCA, w will be the eigenvector corresponding to the maximum eigenvalue of $S_W^{-1} S_B$

LDA – 2 – Class Example (1)




```
% samples for the first class
X1=[5,3;
    3,5;
    3,4;
    4,7;
    5,5];
% samples for the second class
X2=[10,10;
    7,8;
    10,5;
    9,7;
    11,8];
```

Solving for λ in $S_W^{-1} S_B w = \lambda w$

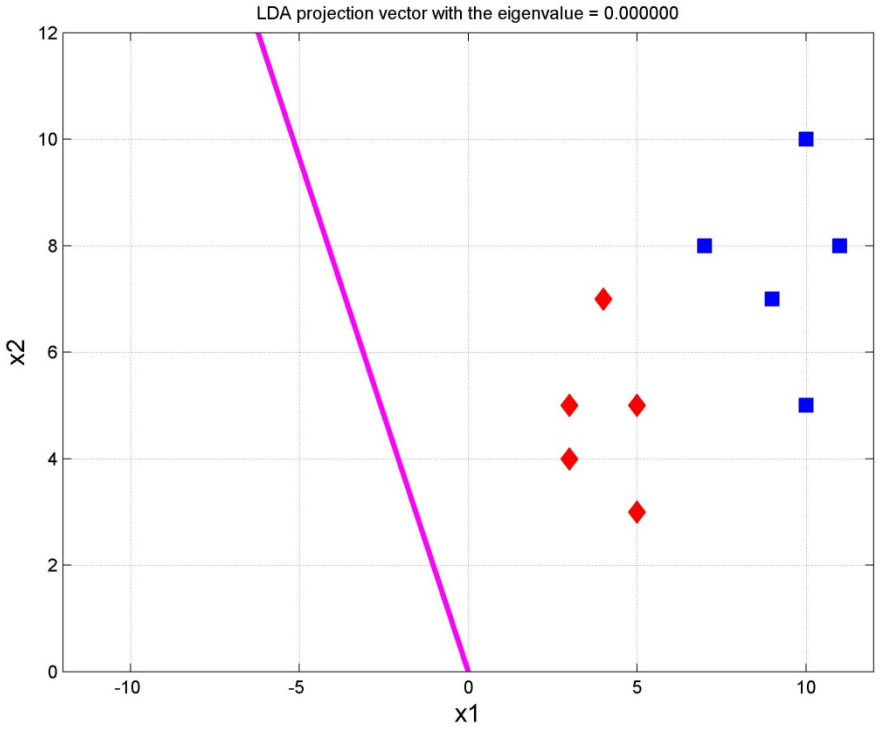
Eigenvalues	Eigenvector
10.8153	(0.9422, 0.3350)'
0	(-0.4603, 0.8878)'

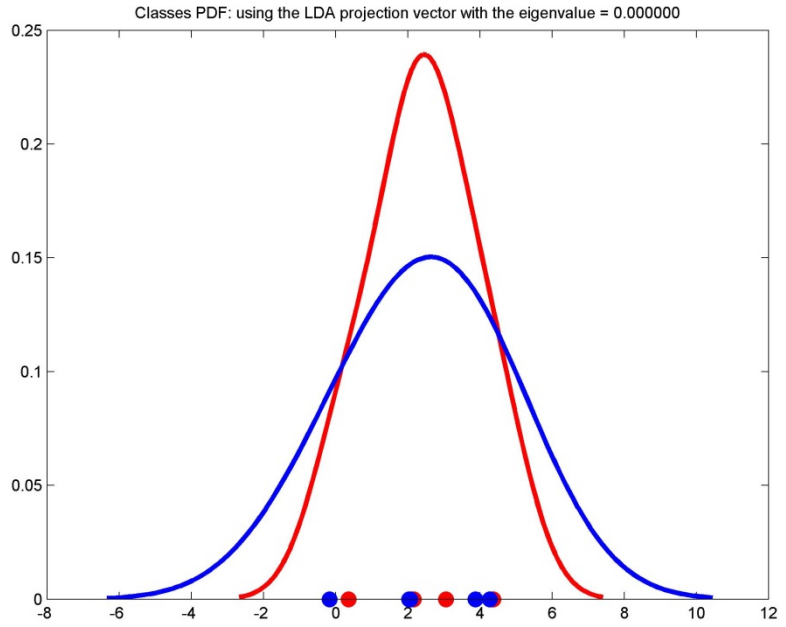
Or directly;

$$w^* = S_W^{-1}(\mu_1 - \mu_2)$$

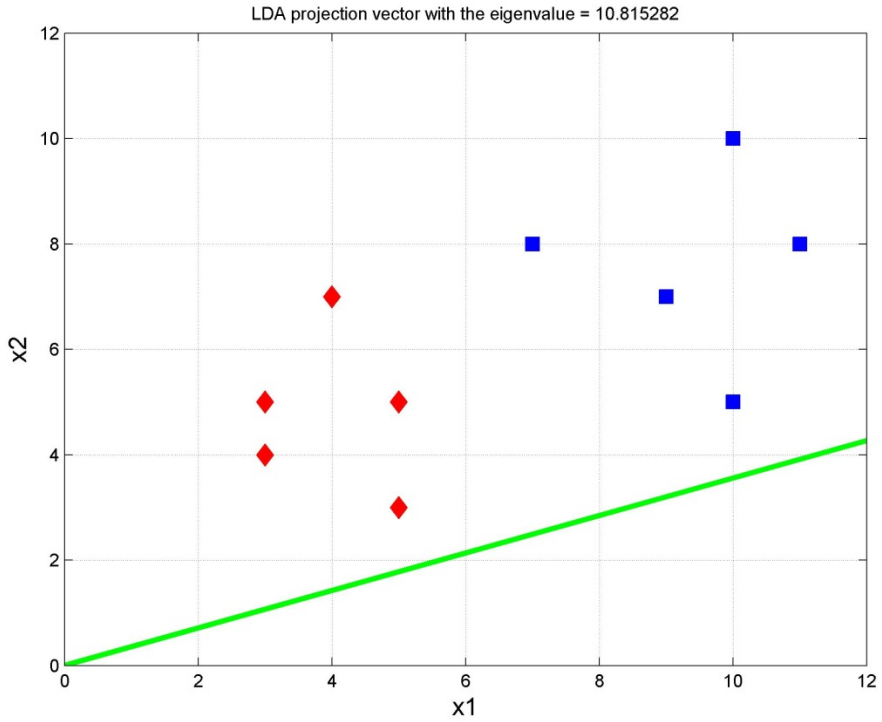
Will result in $w^* = \begin{pmatrix} 0.9422 \\ 0.3350 \end{pmatrix}$

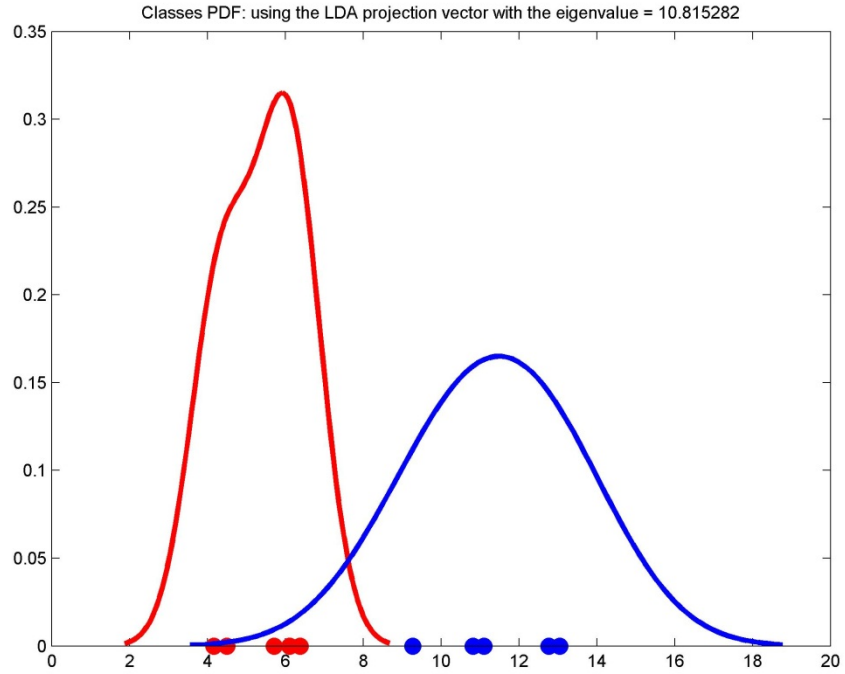
LDA – 2 – Class Example (2)





LDA – 2 – Class Example (3)





LDA – Multiple Classes (1)

- We have $C > 2$ classes.
- Seek $(C-1)$ projections $[z_1, z_2, \ldots, z_{C-1}]$ by means of $(C-1)$ projection vectors w_k
- w_k can be arranged by columns into a projection matrix $W = [w_1|w_2|\ldots|w_{C-1}]$ such that

$$z_k = w_k^T x \Rightarrow z = W^T x$$

LDA – Multiple Classes (2)

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

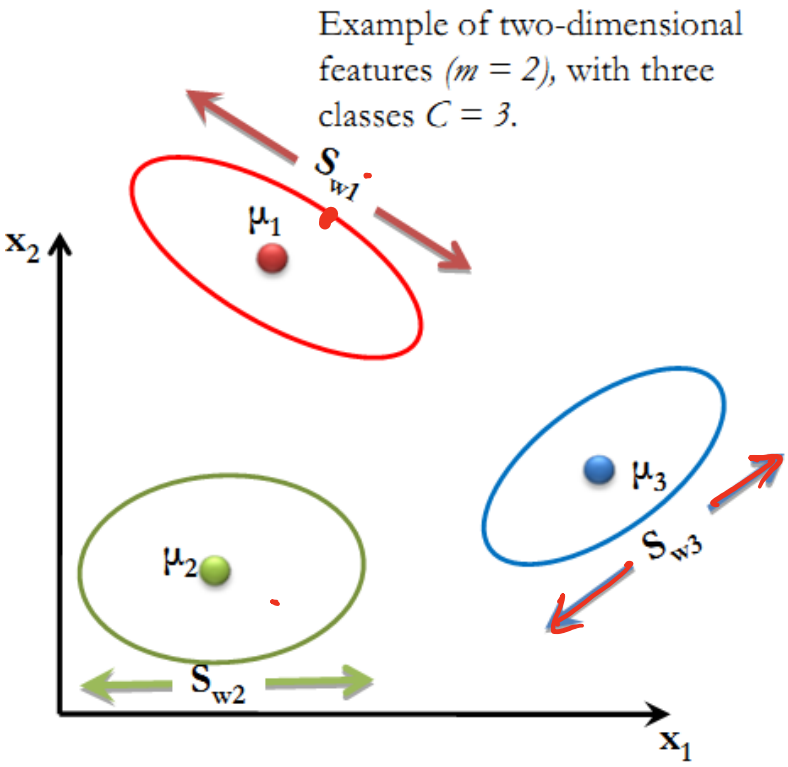
$$S_w = S_1 + S_2$$

- This can be generalized in the C-classes scenario as:

$$S_w = \sum_k S_k$$

Where $S_k = \sum_{x \in C_k} (x - \mu_k)(x - \mu_k)^T$

μ_k is the mean of Class C_k



LDA – Multiple Classes (3)

- Similarly the between class scatter was computed as
  $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$
- For the C- class case, the between-class scatter is computing wrt mean of all classes:

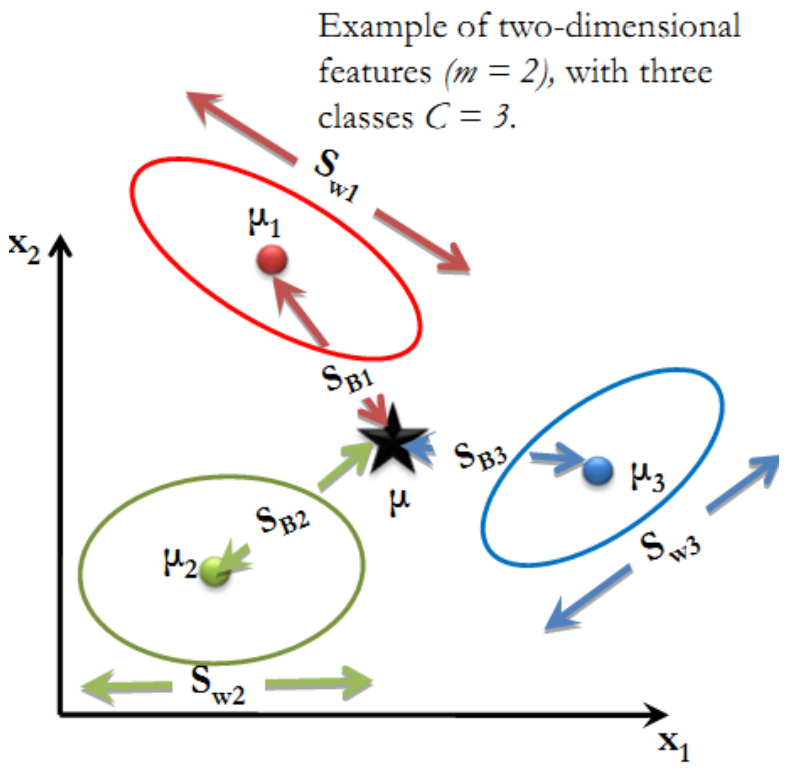
$$S_B = \sum_{k=1}^{C} N_k(\mu_k - \mu)(\mu_k - \mu)^T$$

Where

μ is the mean of all the data points

μ_k is the mean of all the points belonging to class C_k

N_k is the number of points belonging to C_k



LDA – Multiple Classes (4)

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

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

- Seek a projection that *maximizes the ratio of between-class to within-class scatte*r.
- Since the projection is no longer a scalar (it has $C - 1$ dimensions), the determinant of the scatter matrices are used to obtain the scalar objective function:

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

LDA – Multiple Classes (5)

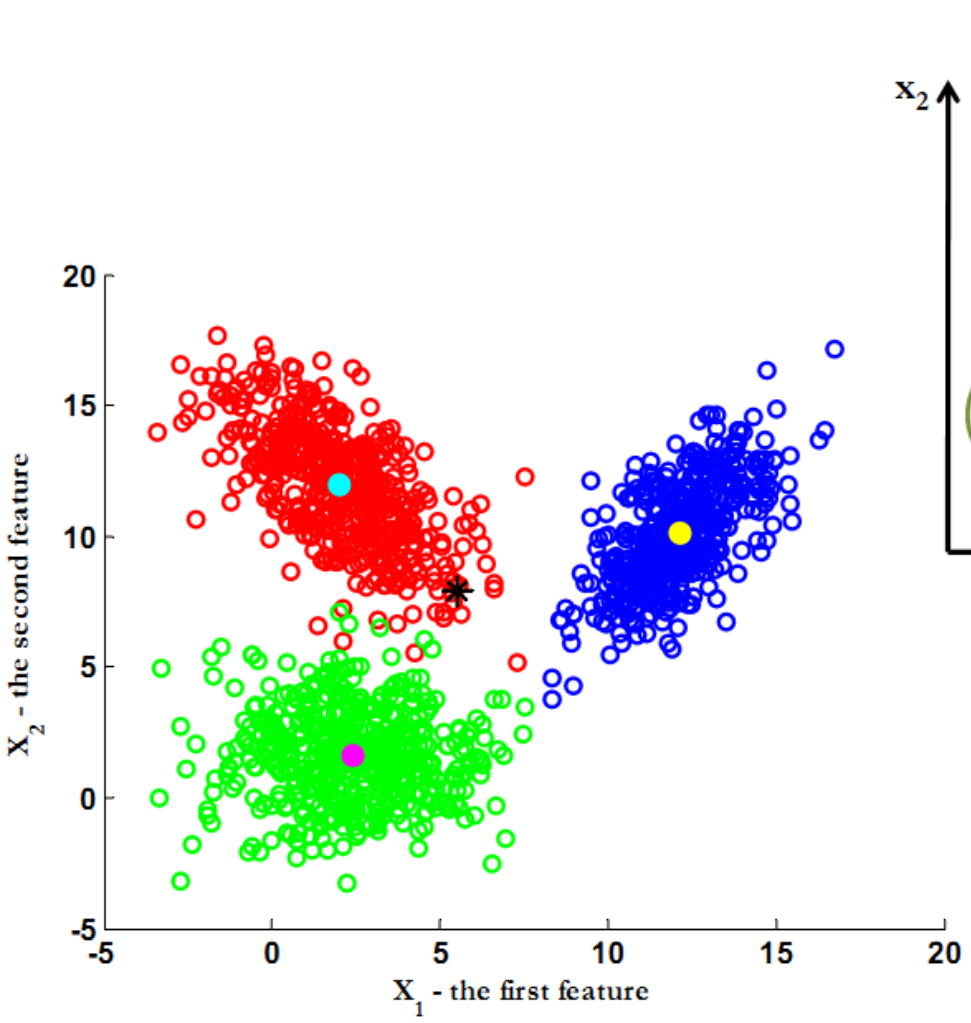
- To find the maximum of $J(W)$, differentiate it wrt W and equate it to 0
- For the C-class case, we have $C-1$ projection vectors, hence the eigenvalue problem can be generalized to the C-classes case as:

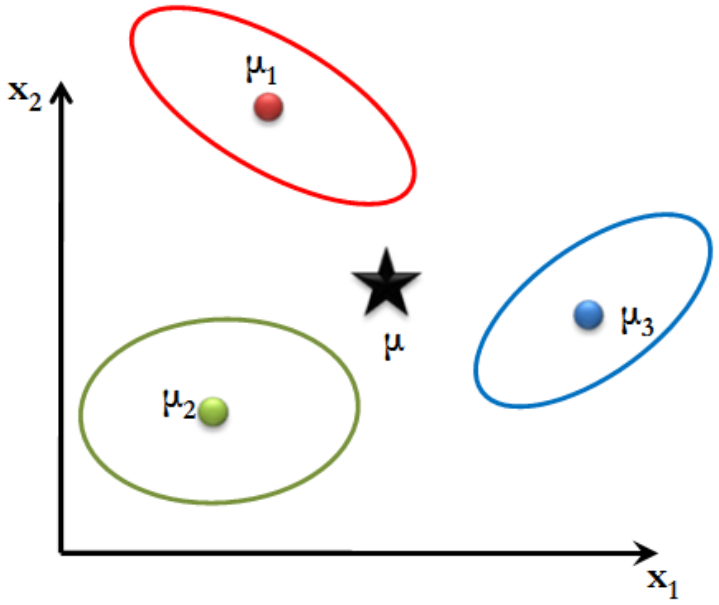
$$S_W^{-1} S_B w_k = \lambda_k w_k$$

- It can be thus shown that the optimal projection matrix W^* is the one whose columns are the eigenvectors corresponding to the largest eigenvalues of the following generalized eigenvalue problem:

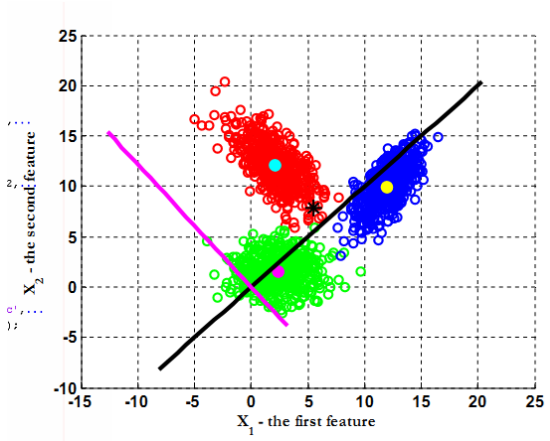
$$S_W^{-1} S_B W^* = \lambda W^*$$

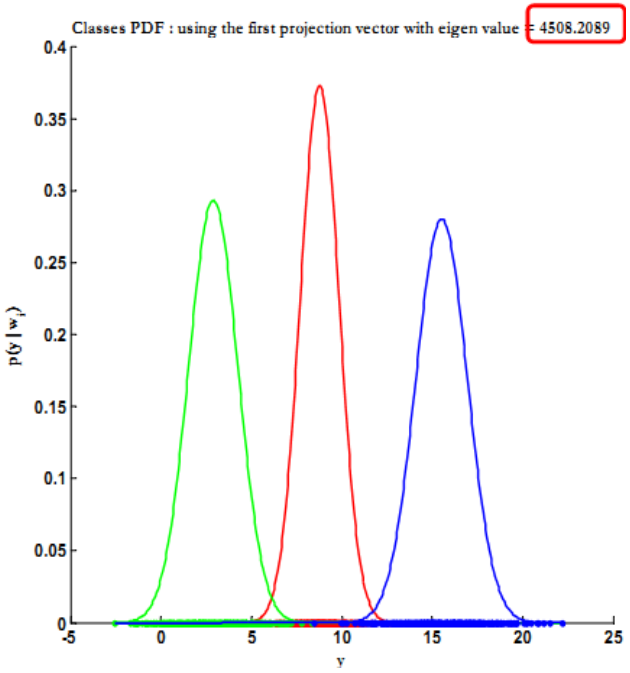
LDA – Multiple Class – Example (1)

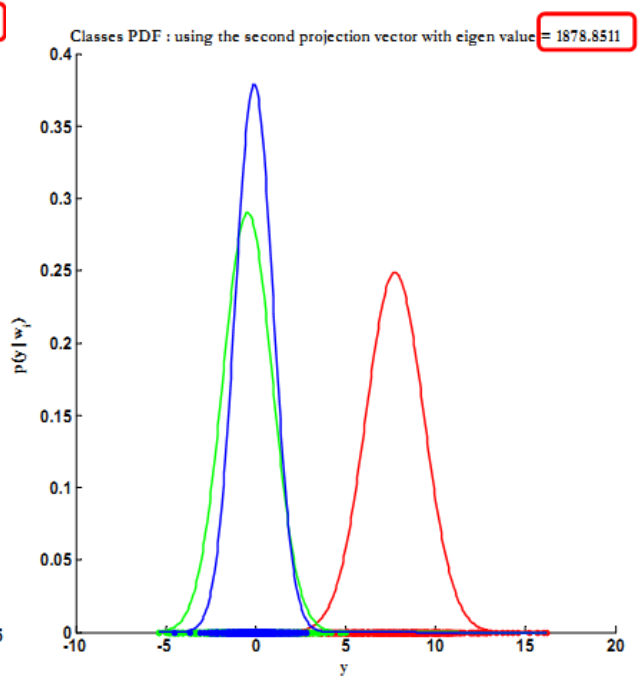




LDA – Multiple Class – Example (2)

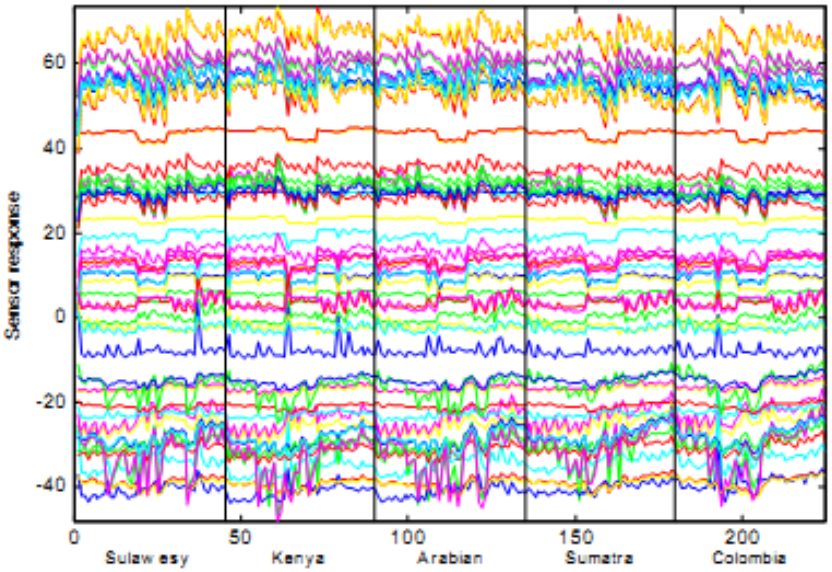


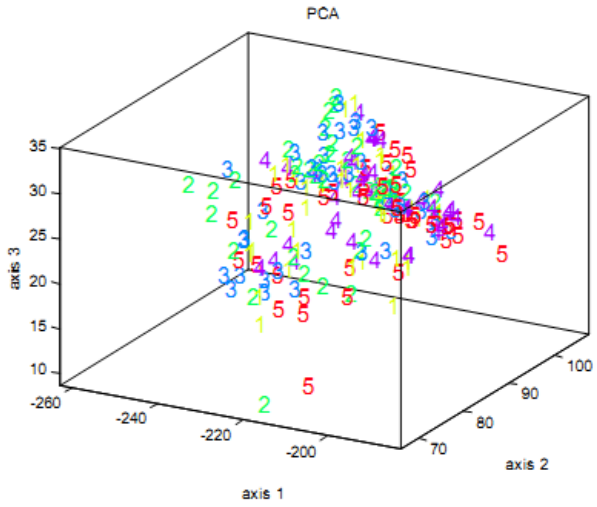


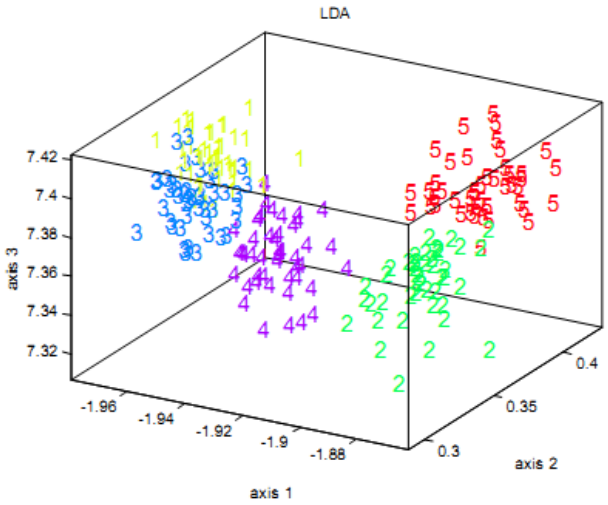


PCA vs LDA

- Coffee Bean dataset
  - Five types of coffee beans were presented to an array of chemical gas sensors
  - For each coffee type, 45 “sniffs” were performed and the response of the gas sensor array was recorded in order to obtain a 60 dimensional feature vector

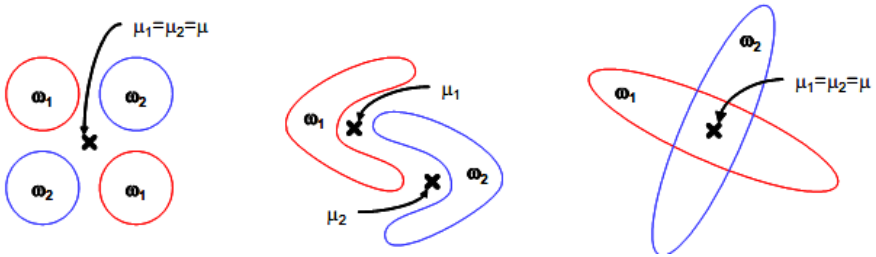




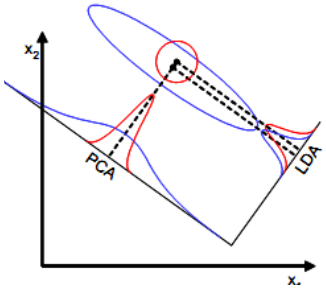


Limitations of LDA

- LDA produces at most C-1 feature projections
  - If the classification error estimates establish that more features are needed, some other method must be employed to provide those additional features
- 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 that may be needed for classification



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



Summary

- Motivation
- Unsupervised Dimensionality Reduction
  - Principal Component Analysis
- Supervised Dimensionality Reduction
  - Linear Discriminant Analysis