Pattern Recognition

Feature Selection and Reduction

Motivation

- In practical multi-category applications, it is not unusual to encounter problems involving tens or hundreds of features.

- Increasing the number of features increases the complexity of the classifier.

Curse of Dimensionality

- It has frequently been observed in practice that, beyond a certain point, adding new features leads to worse rather than better performance.
- The fact that the convergence of any estimator to the true value of a smooth function defined on a space of high dimension is very slow after a point is called the curse of dimensionality

Optimum Dimensionality

- The aim of optimizing dimensionality of the feature space is:
  - Finding the optimum number of features for best performance (feature reduction)
  - Choosing the most suitable features for best discrimination (feature selection)

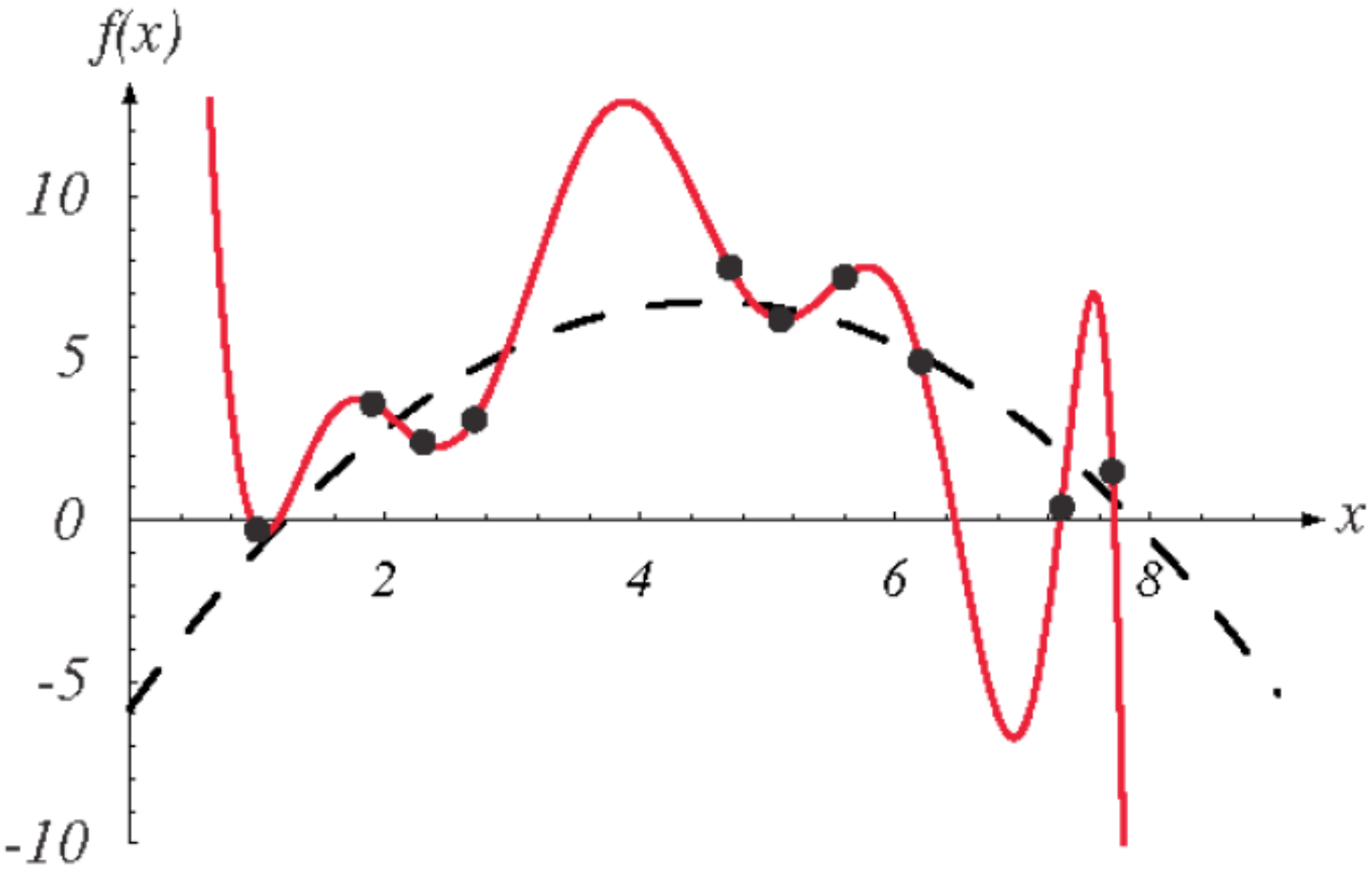
Considerations for Optimum Dimensionality

- There are two issues that we must be careful about:
  - How is the classification accuracy affected by the dimensionality (relative to the amount of training data)?
  - How is the computational complexity of the classifier affected by the dimensionality?

Problems of Dimensionality

- Potential reasons for increase in error include
  - wrong assumptions in model selection,
  - estimation errors due to the finite number of training samples for high-dimensional observations (overfitting).
- Potential solutions include
  - reducing the dimensionality,
  - simplifying the estimation.

Example: Overfitting



Reducing Dimensionality

- Dimensionality can be reduced by
  - redesigning the features,
  - selecting an appropriate subset among the existing features,
  - combining existing features.

Issues in feature reduction

- Linear vs. non-linear transformations.
- Use of class labels or not (depends on the availability of training data).
- Training objective:
  - minimizing classification error (discriminative training),
  - minimizing reconstruction error (PCA),
  - maximizing class separability (LDA),
  - retaining interesting directions (projection pursuit),
  - making features as independent as possible (ICA).

Feature Reduction

- Given $x \in R^d$, the goal is to find a linear transformation A that gives $y = A^T x \in R^{d'}$ where $d' < d$
- Two classical approaches for finding optimal linear transformations are:
  - Principal Components Analysis (PCA): Seeks a projection that best represents the data in a least-squares sense.
  - Linear Discriminant Analysis (LDA): Seeks a projection that best separates the data in a least-squares sense

Principal Component Analysis (PCA)

- Given $x_1, \ldots, x_n \in R^d$, the goal is to find a d'-dimensional subspace where the reconstruction error of x_i in this subspace is minimized

PCA

- Let x be an N-dimensional random vector, represented as a linear combination of orthonormal basis vectors $[\varphi_1 | \varphi_2 | \ldots | \varphi_N]$ as

$$x = \sum_{i=1}^{N} y_i \varphi_i \quad \text{where} \quad \varphi_i \,|\, \varphi_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

- Suppose we choose to represent x with only M (M<N) of the basis vectors. We can do this by replacing the components $[y_{M+1}, \ldots, y_N]^T$ with some pre-selected constants b_i

$$\hat{x}(M) = \sum_{i=1}^{M} y_i \varphi_i + \sum_{i=M+1}^{N} b_i \varphi_i$$

- The representation error is then

$$\Delta x(M) = x - \hat{x}(M) = \sum_{i=1}^{N} y_i \varphi_i - \left(\sum_{i=1}^{M} y_i \varphi_i + \sum_{i=M+1}^{N} b_i \varphi_i \right) = \sum_{i=M+1}^{M} (y_i - b_i) \varphi_i$$

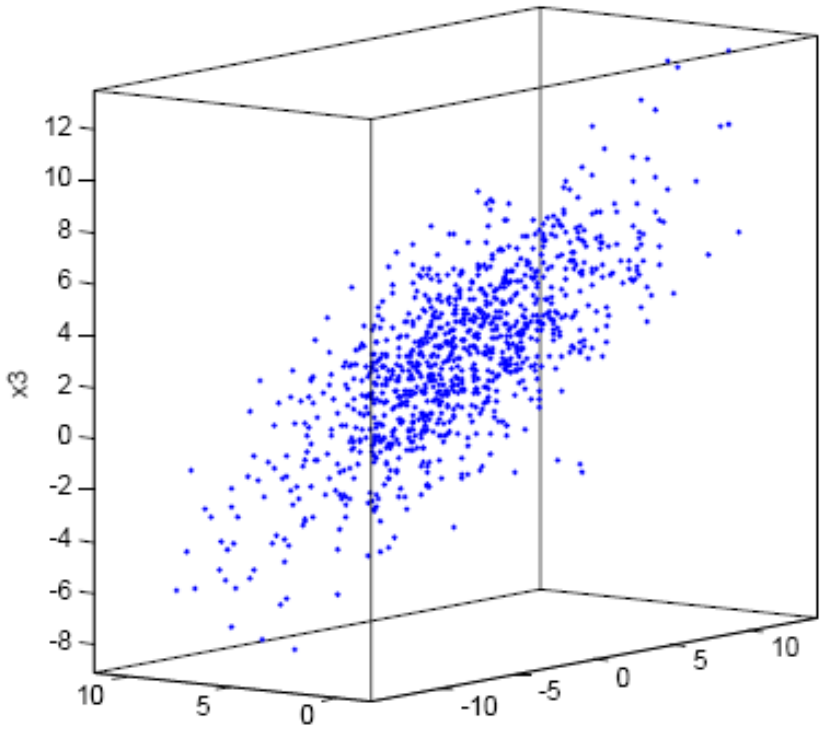
PCA

- We can measure this representation error by the mean-squared magnitude of Δx
- Our goal is to find the basis vectors φ_i and constants b_i that minimize this mean-square error

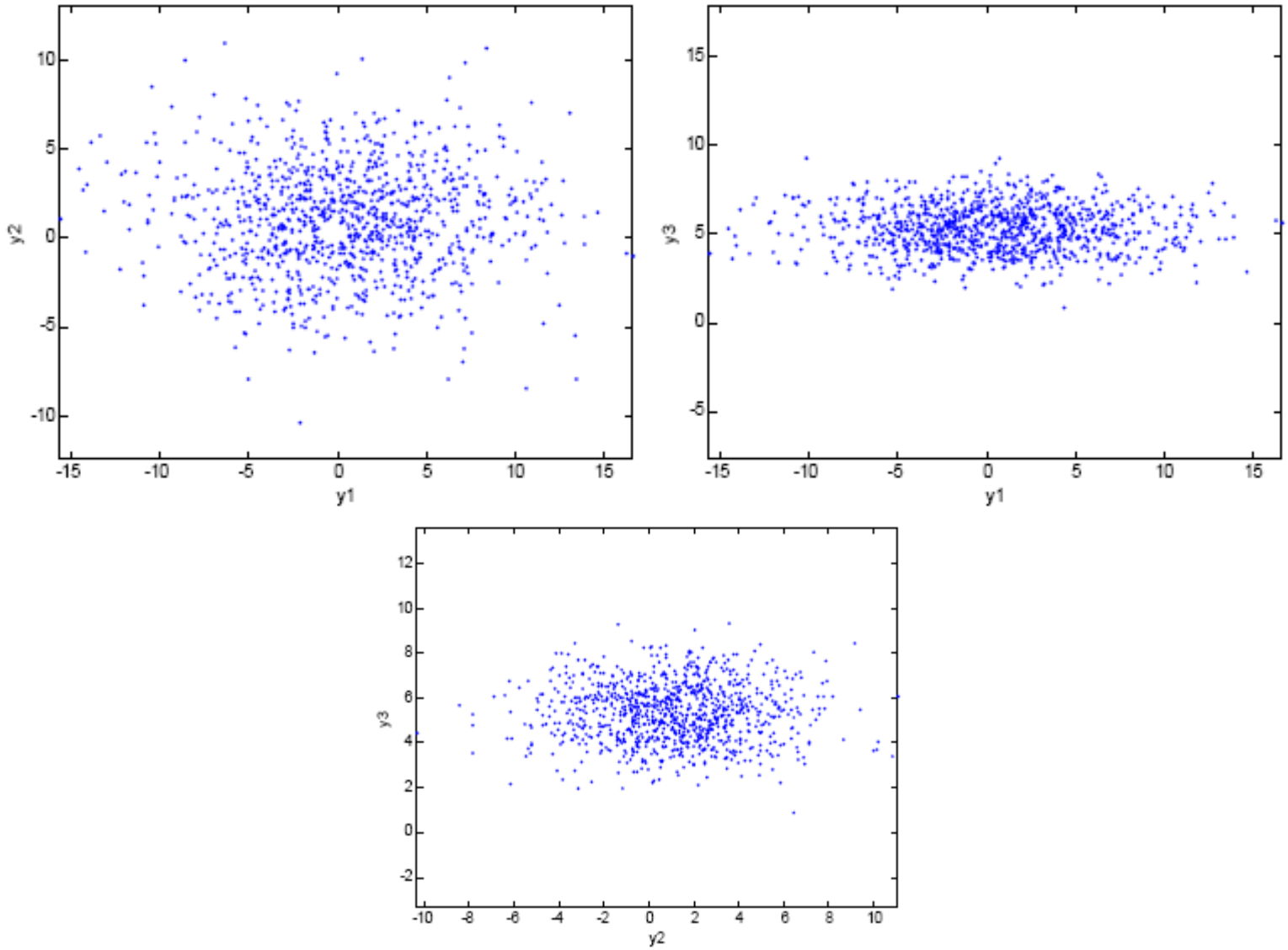
$$\bar{\varepsilon}^2(M) = E\left[\left|\Delta x(M)\right|^2\right] = E\left[\sum_{i=M+1}^{N}\sum_{j=M+1}^{N}(y_i - b_i)(y_j - b_j)\varphi_i^T\varphi_j\right] = \sum_{i=M+1}^{N}E\left[(y_i - b_i)^2\right]$$

PCA Example

$$\mu = [0 \;\; 5 \;\; 2]^T \text{ and } \Sigma = \begin{bmatrix} 25 & -1 & 7 \\ -1 & 4 & -4 \\ 7 & -4 & 10 \end{bmatrix}$$



Principal Component Projections

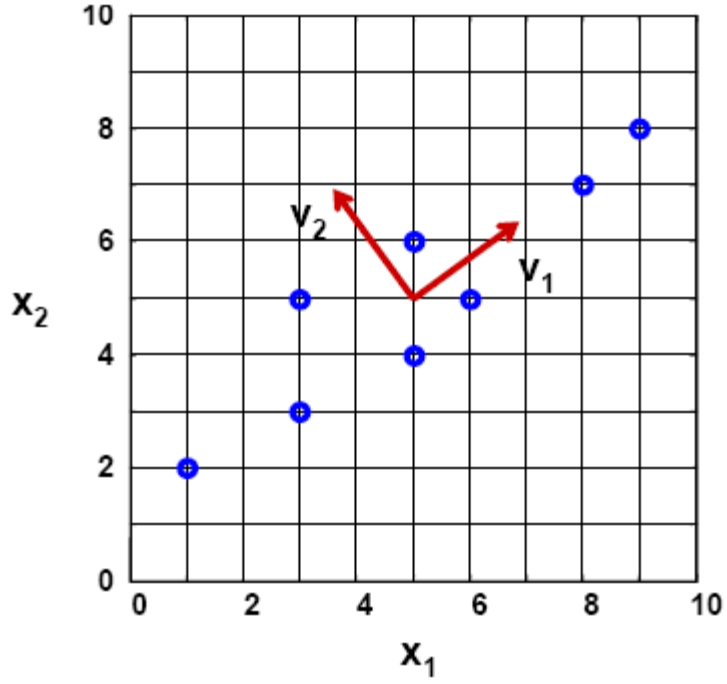


Example (PCA)

$X=(x_1,x_2)=\{(1,2),(3,3),(3,5),(5,4),(5,6),(6,5),(8,7),(9,8)\}$

The (biased) covariance estimate of the data is:

$$\Sigma_x = \begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$



The eigenvalues are the zeros of the characteristic equation

$$\Sigma_x v = \lambda v \Rightarrow |\Sigma_x - \lambda I| = 0 \Rightarrow \begin{vmatrix} 6.25-\lambda & 4.25 \\ 4.25 & 3.5-\lambda \end{vmatrix} = 0 \Rightarrow \lambda_1 = 9.34;\ \lambda_2 = 0.41;$$

Example (PCA)

The eigenvectors are the solutions of the system

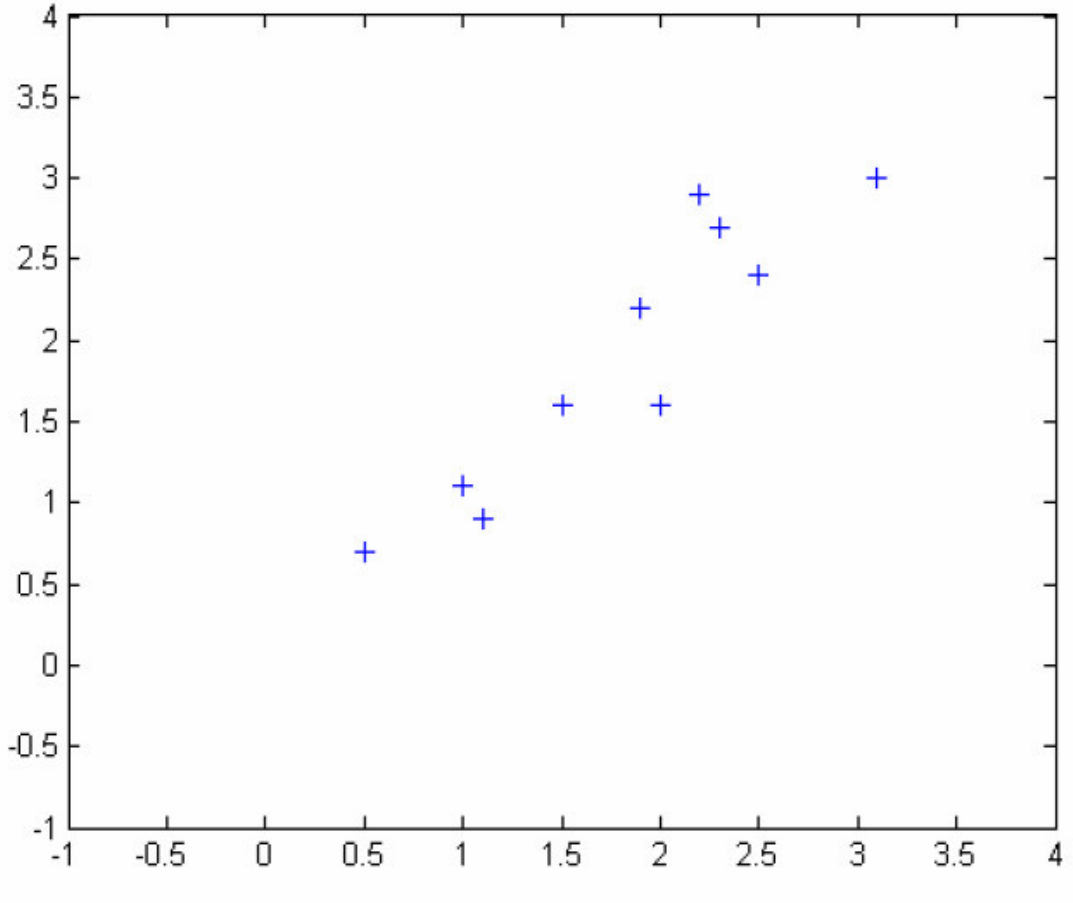
$$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}\begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} \lambda_1 v_{11} \\ \lambda_1 v_{12} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0.81 \\ 0.59 \end{bmatrix}$$

$$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}\begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} \lambda_2 v_{21} \\ \lambda_2 v_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} -0.59 \\ 0.81 \end{bmatrix}$$

Example (PCA)

Data =

2.5000	2.4000
0.5000	0.7000
2.2000	2.9000
1.9000	2.2000
3.1000	3.0000
2.3000	2.7000
2.0000	1.6000
1.0000	1.1000
1.5000	1.6000
1.1000	0.9000



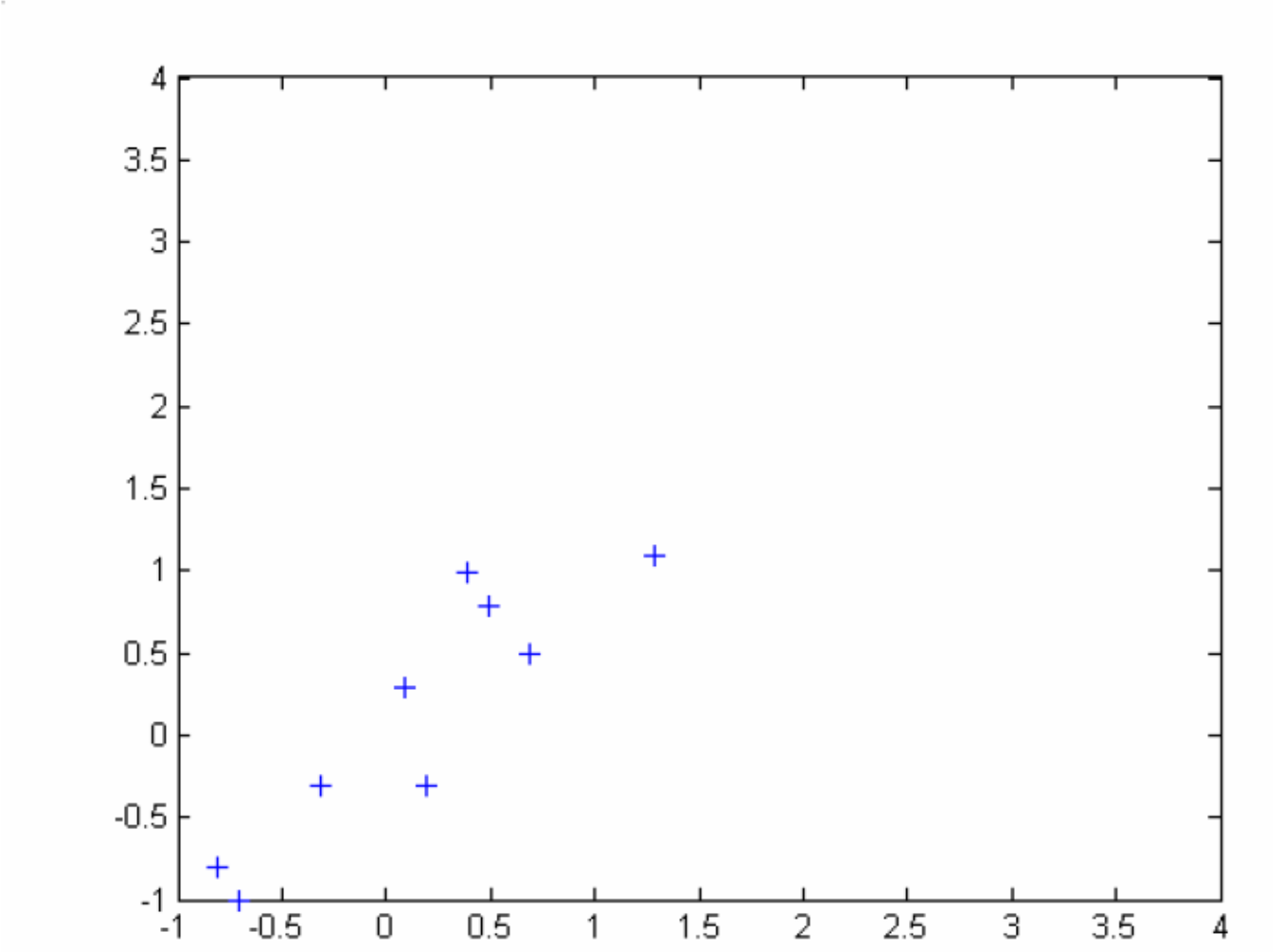
Example (Cont.)

$\hat{Data}$ = Data-repmat(mean(Data),size(Data,1),1)

0.6900	0.4900
-1.3100	-1.2100
0.3900	0.9900
0.0900	0.2900
1.2900	1.0900
0.4900	0.7900
0.1900	-0.3100
-0.8100	-0.8100
-0.3100	-0.3100
-0.7100	-1.0100

Example(Cont.)

$\hat{Data}$



Example(Cont.)

Covaraince Matrix

$$\begin{bmatrix} 0.6166 & 0.6154 \\ 0.6154 & 0.7166 \end{bmatrix}$$

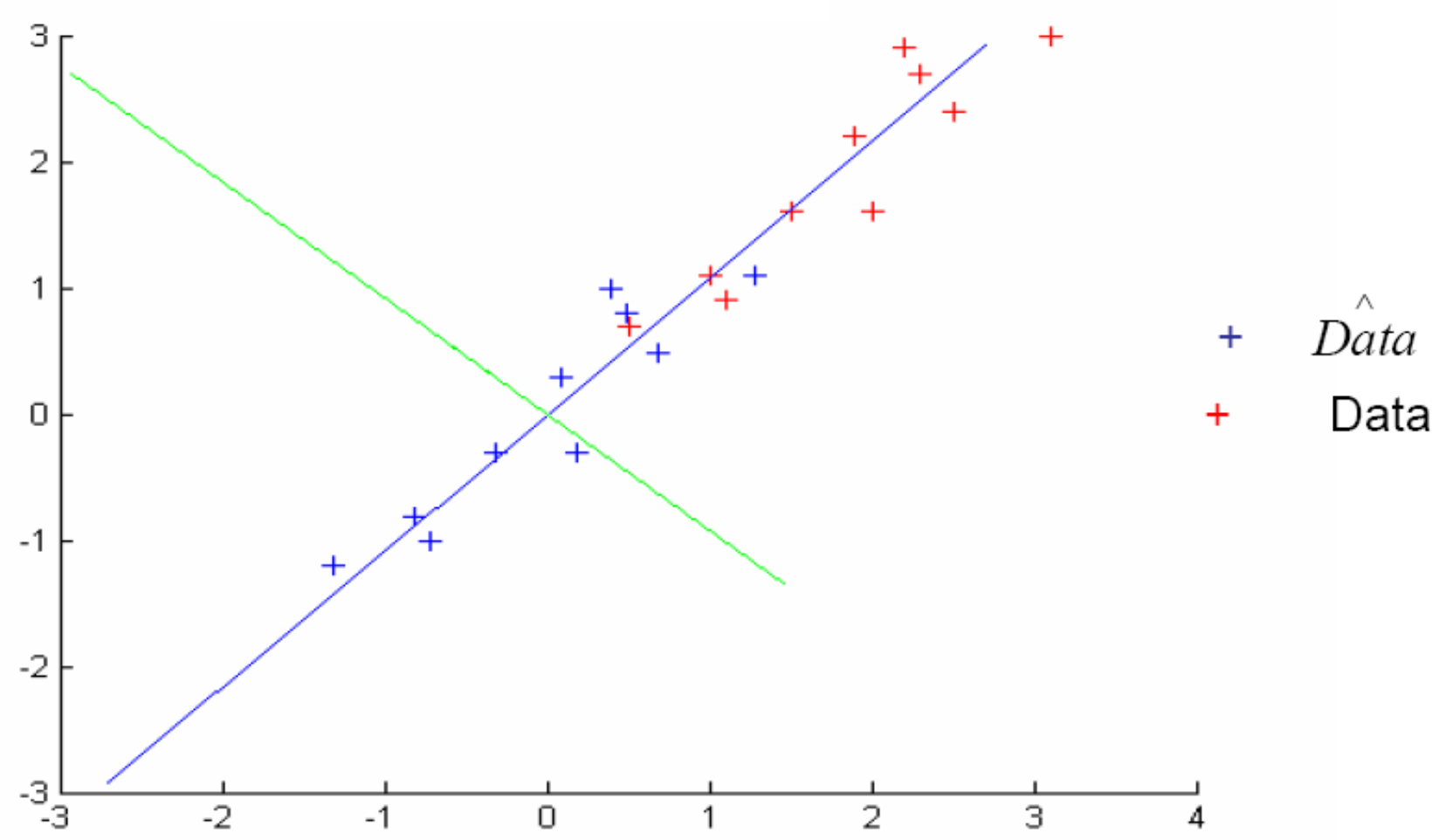
Eigenvectors

$$\begin{bmatrix} -0.7352 & 0.6779 \\ 0.6779 & 0.7352 \end{bmatrix}$$

Eigenvalues

$$\begin{bmatrix} 0.0491 \\ 1.2840 \end{bmatrix}$$

Example(Cont.)



Example(Cont.)

Eigenvector with the “Highest” eigenvalue is the “principle component” of the data set

Principle component from the example

-0.7352	**0.6779**	**0.0491**
0.6779	**0.7352**	**1.2840**

Feature Vectors (p vectors)

– Selected from n eigenvectors (p<n)

Example (Cont.)

Deriving new data set

$$Y = XP$$

Where Y is the data set, which are translated into new axes. Note that the matrix P represents the new axes. X is the data set represented N by d

N is the number of data and d is the number of dimensions.

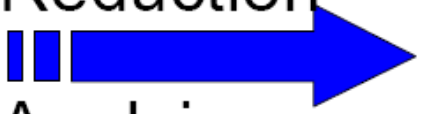
P is the eigenvectors with column vectors arranged in the order from the highest eigenvalues to lowest eigenvalues.

Example (Cont.)

Transformed Data (Y) =

-0.8280	-0.1751
1.7776	0.1429
-0.9922	0.3844
-0.2742	0.1304
-1.6759	-0.2095
-0.9130	0.1753
0.0991	-0.3498
1.1446	0.0464
0.4381	0.0178
1.2239	-0.1627

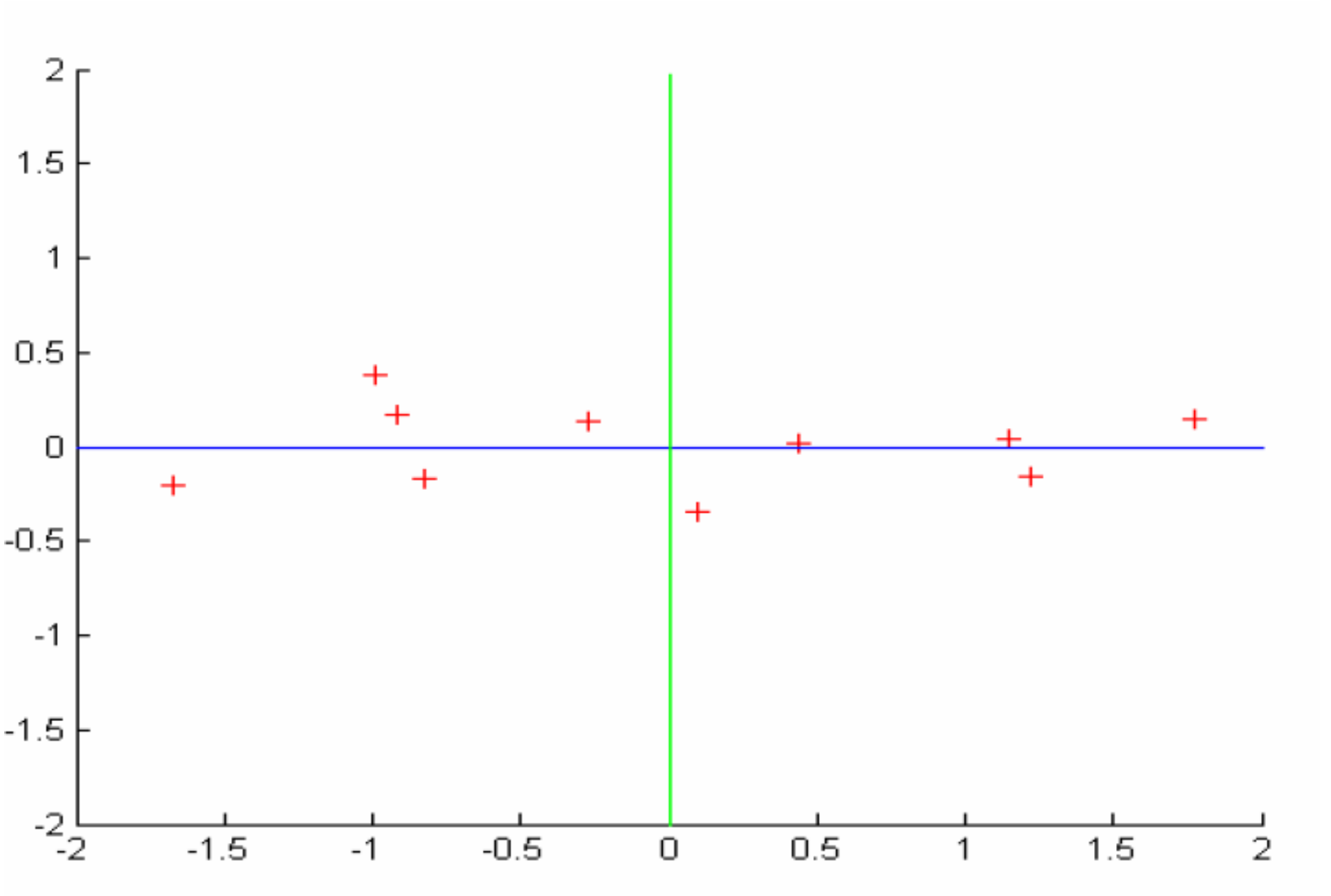
Reduction



Applying major component

-0.8280
1.7776
-0.9922
-0.2742
-1.6759
-0.9130
0.0991
1.1446
0.4381
1.2239

Example (Cont.)



Example (Cont.)

- To recover data from Y
  - $Y = XP$
  - $X = YP^{-1}$, $P^{-1} = P^{t}$, notice that X is centered according to the Mean of Data

Recovering Data

$Y =$

$$\begin{bmatrix} -0.8280 \\ 1.7776 \\ -0.9922 \\ -0.2742 \\ -1.6759 \\ -0.9130 \\ 0.0991 \\ 1.1446 \\ 0.4381 \\ 1.2239 \end{bmatrix}$$

$YP^{-1} + \text{Mean(Data)} =$

$$\begin{bmatrix} 2.3713 & 2.5187 \\ 0.6049 & 0.6031 \\ 2.4826 & 2.6395 \\ 1.9959 & 2.1116 \\ 2.9461 & 3.1421 \\ 2.4289 & 2.5812 \\ 1.7428 & 1.8371 \\ 1.0341 & 1.0685 \\ 1.5130 & 1.5879 \\ 0.9803 & 1.0102 \end{bmatrix}$$

Linear Discriminant Analysis

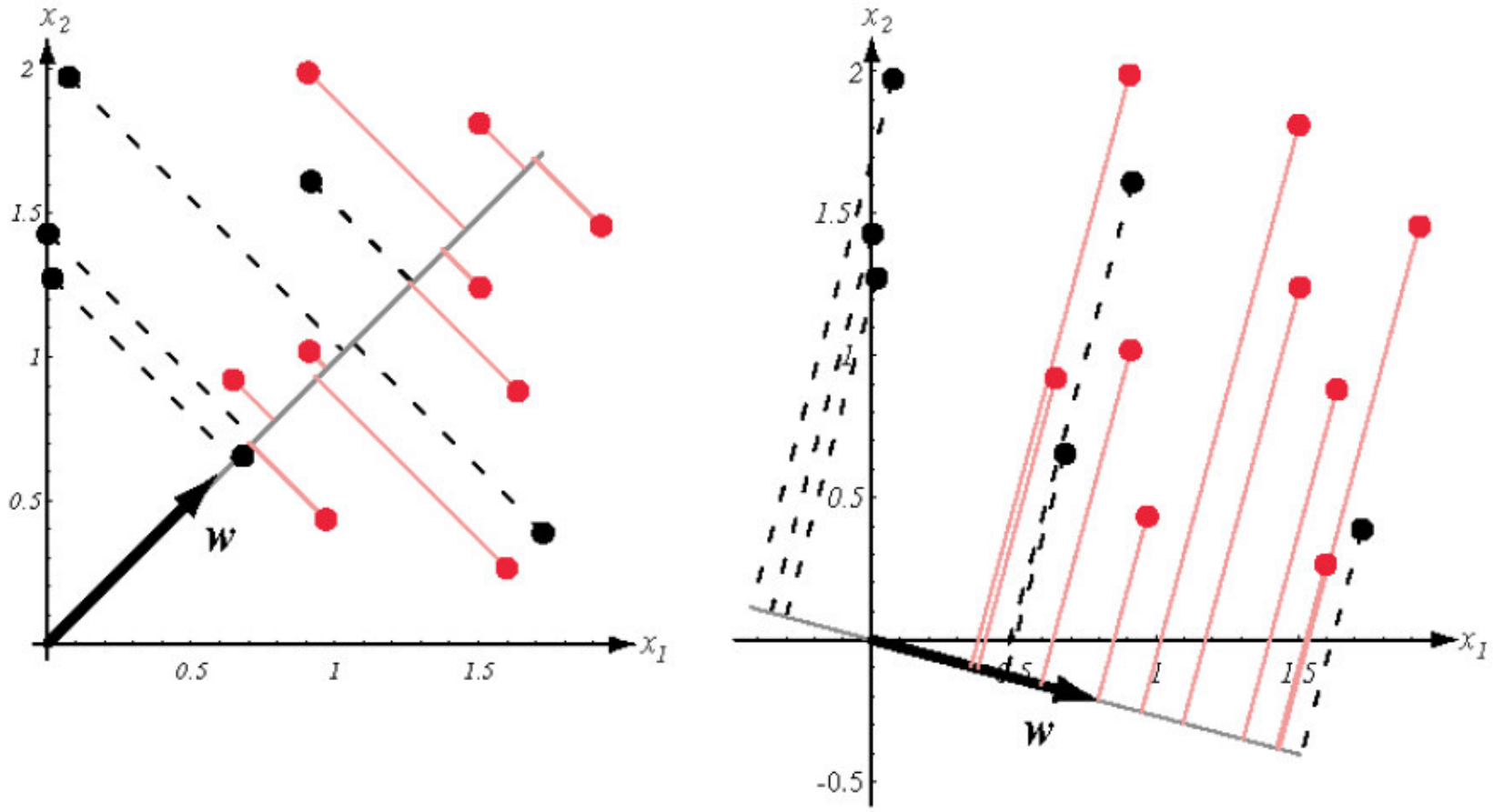
Whereas PCA seeks directions that are efficient for representation, discriminant analysis seeks directions that are efficient for discrimination.

Given $\mathbf{x_1}, \ldots, \mathbf{x_n} \in \mathbb{R}^d$ divided into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$ corresponding to the classes w_1 and w_2, respectively, the goal is to find a projection onto a line defined as

$$y = \mathbf{w}^T \mathbf{x}$$

where the points corresponding to $\mathcal{D}_1$ and $\mathcal{D}_2$ are well separated.

Linear Discriminant Analysis



Linear Discriminant Analysis

The criterion function for the best separation can be defined as

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

where $\tilde{m}_i = \frac{1}{\#\mathcal{D}_i} \sum_{y \in w_i} y$ is the sample mean and $\tilde{s}_i^2 = \sum_{y \in w_i} (y - \tilde{m}_i)^2$ is the scatter for the projected samples labeled w_i.

Linear Discriminant Analysis

This is called the *Fisher's linear discriminant* with the geometric interpretation that the best projection makes the difference between the means as large as possible relative to the variance.

Linear Discriminant Analysis

To compute the optimal $\mathbf{w}$, we define the *scatter matrices* $\mathbf{S}_i$

$$\mathbf{S}_i = \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T \text{ where } \mathbf{m}_i = \frac{1}{\#\mathcal{D}_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x},$$

the *within-class scatter matrix* $\mathbf{S}_\mathbf{W}$

$$\mathbf{S}_\mathbf{W} = \mathbf{S}_1 + \mathbf{S}_2,$$

and the *between-class scatter matrix* $\mathbf{S}_\mathbf{B}$

$$\mathbf{S}_\mathbf{B} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T.$$

Linear Discriminant Analysis

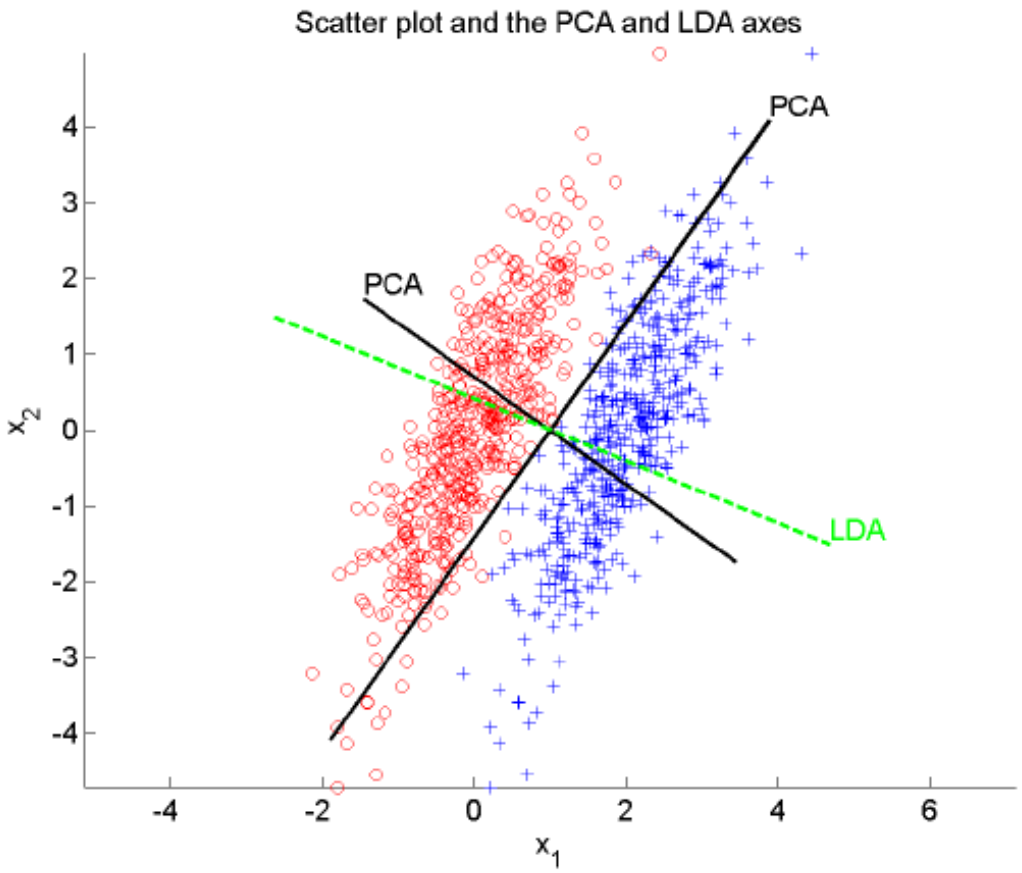
Then, the criterion function becomes

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S_B} \mathbf{w}}{\mathbf{w}^T \mathbf{S_W} \mathbf{w}}$$

and the optimal $\mathbf{w}$ can be computed as

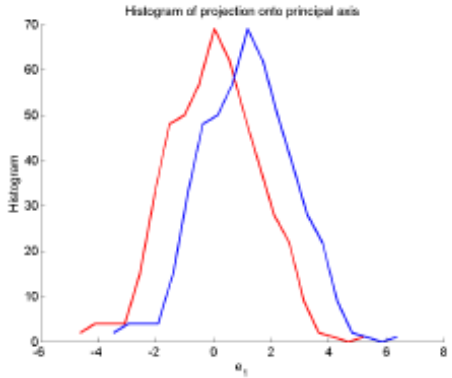
$$\mathbf{w} = \mathbf{S_W^{-1}}(\mathbf{m_1} - \mathbf{m_2}).$$

Example

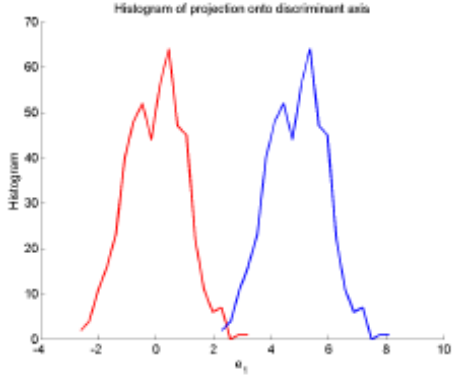


(a) Scatter plot.

Example

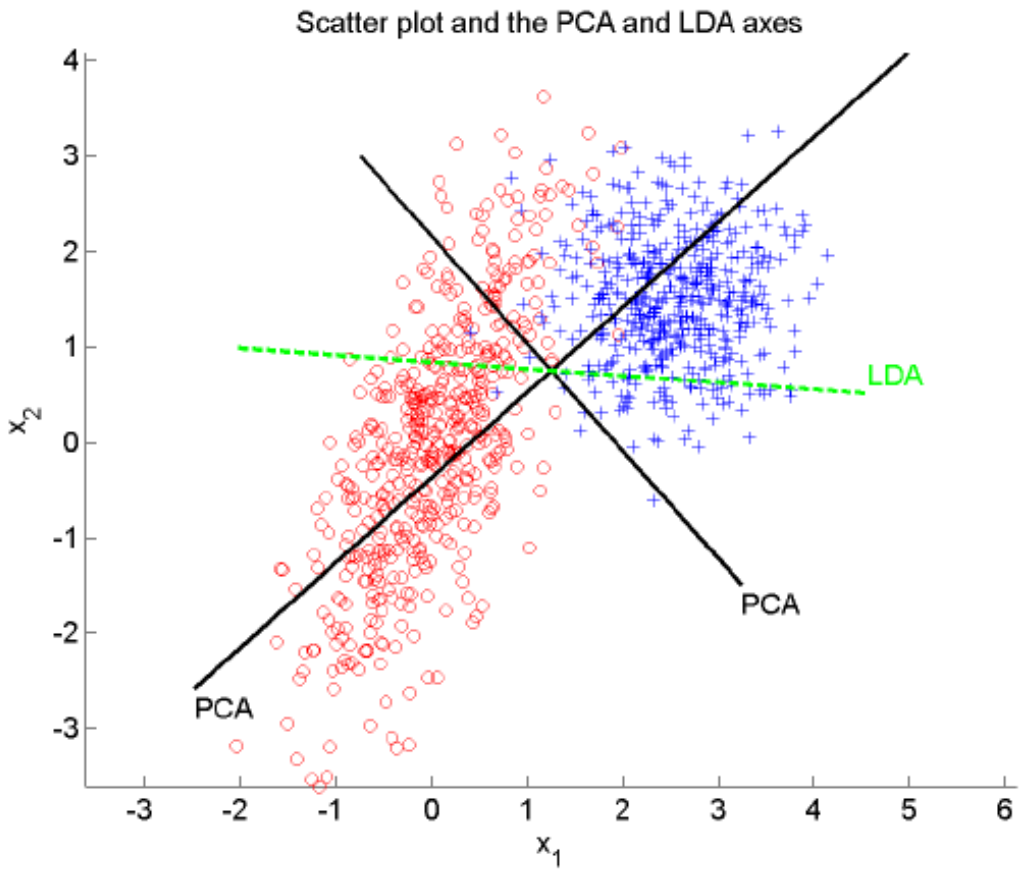


(b) Projection onto the first PCA axis.



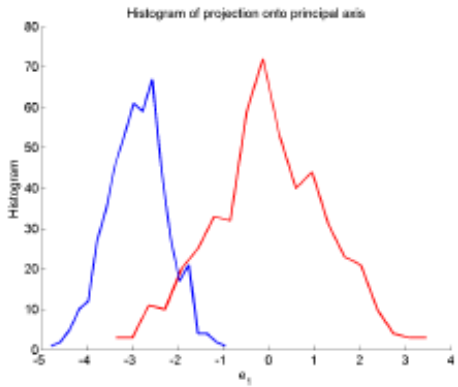
(c) Projection onto the first LDA axis.

Example

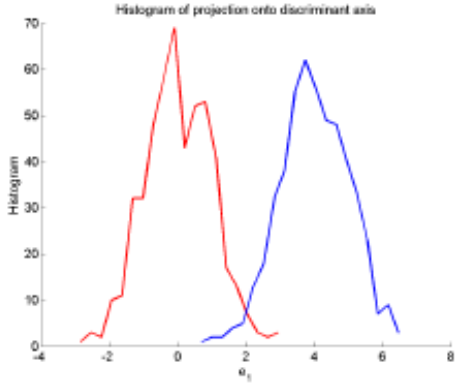


(a) Scatter plot.

Example



(b) Projection onto the first PCA axis.



(c) Projection onto the first LDA axis.

Potential Term Project

- Assume each face image is given as a 2D matrix of integer numbers ranging from 0 to 255. (Each value represents a gray level, o being black, 255 white and all other numbers in between gray shadings). Also assume images of n persons at different conditions have been given (image of each person). Convert each matrix into a vector and use PCA to classify the images and identify a person if a new image is provided.
- Use public face image databases for training and testing your algorithm.

Sample Images

Questions?