Feature Extraction (PCA & LDA)

CE-725: Statistical Pattern Recognition
Sharif University of Technology
Spring 2013

Soleymani

Outline

- What is feature extraction?
- Feature extraction algorithms
  - Linear Methods
    - Unsupervised: Principal Component Analysis (PCA)
      - Also known as Karhonen-Loeve (KL) transform
    - Supervised: Linear Discriminant Analysis (LDA)
      - Also known as Fisher's Discriminant Analysis (FDA)

Dimensionality Reduction: Feature Selection vs. Feature Extraction

- Feature **selection**
  - Select a subset of a given feature set
- Feature **extraction** (e.g., PCA, LDA)
  - A linear or non-linear transform on the original feature space

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \to \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_{d'}} \end{bmatrix}$$

Feature Selection ($d' < d$)

$$\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \to \begin{bmatrix} y_1 \\ \vdots \\ y_{d'} \end{bmatrix} = f\left(\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}\right)$$

Feature Extraction

Feature Extraction

- Mapping of the original data onto a lower-dimensional space
  - Criterion for feature extraction can be different based on problem settings
    - Unsupervised task: minimize the information loss (reconstruction error)
    - Supervised task: maximize the class discrimination on the projected space

- In the previous lecture, we talked about feature selection:
  - Feature selection can be considered as a special form of feature extraction (only a subset of the original features are used).
  - Example:

$$\mathbf{X}' = \mathbf{X} \times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{X} \in \mathbb{R}^{N \times 4} \qquad \mathbf{X}' \in \mathbb{R}^{N \times 2}$$

Second and thirth features are selected

Feature Extraction

- Unsupervised feature extraction:

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

Feature Extraction

A mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
Or
only the transformed data

$$\boldsymbol{X}' = \begin{bmatrix} {x'}_1^{(1)} & \cdots & {x'}_{d'}^{(1)} \\ \vdots & \ddots & \vdots \\ {x'}_1^{(N)} & \cdots & {x'}_{d'}^{(N)} \end{bmatrix}$$

- Supervised feature extraction:

$$\boldsymbol{X} = \begin{bmatrix} x_1^{(1)} & \cdots & x_d^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_d^{(N)} \end{bmatrix}$$

$$Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{bmatrix}$$

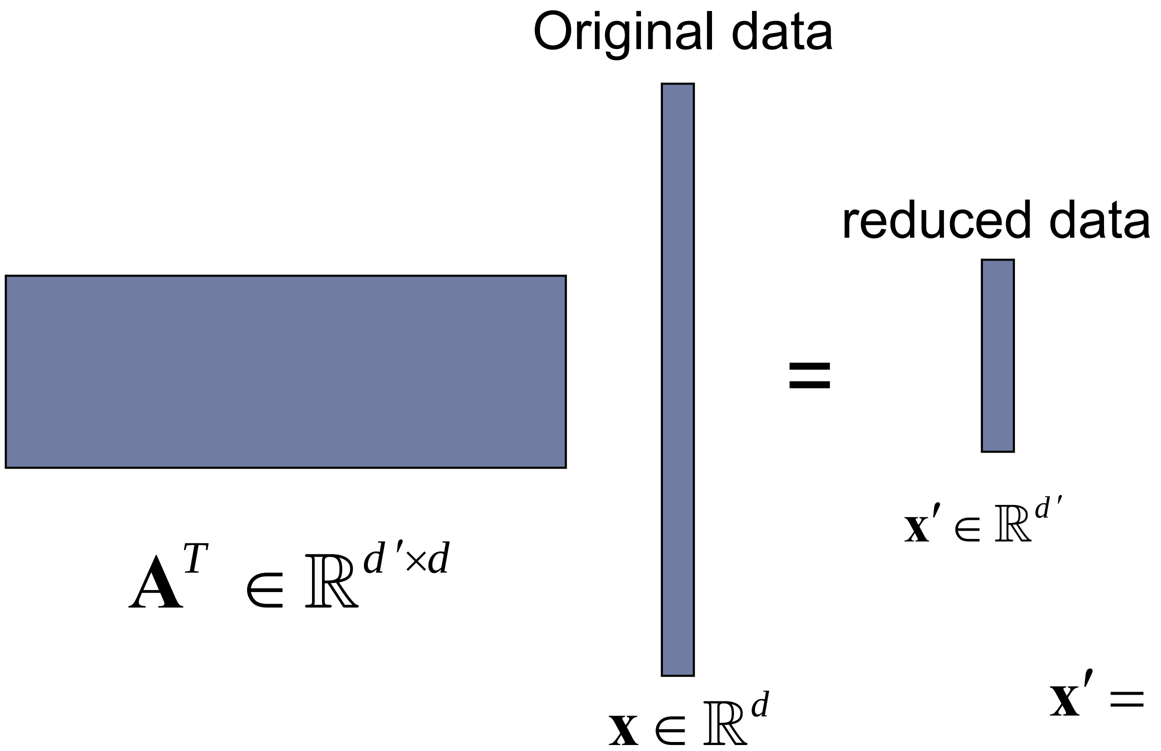
Feature Extraction

A mapping $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$
Or
only the transformed data

$$\boldsymbol{X}' = \begin{bmatrix} {x'}_1^{(1)} & \cdots & {x'}_{d'}^{(1)} \\ \vdots & \ddots & \vdots \\ {x'}_1^{(N)} & \cdots & {x'}_{d'}^{(N)} \end{bmatrix}$$

Linear Transformation

- For linear transformation, we find an explicit mapping $f(\boldsymbol{x}) = \boldsymbol{A}^T \boldsymbol{x}$ that can transform also new data vectors.



$$\mathbf{x}' = \mathbf{A}^T \mathbf{x} \; (d' < d)$$

Linear Transformation

- Linear transformation are simple mappings

$$\mathbf{x}' = \mathbf{A}^T \mathbf{x} \quad (\mathbf{x}'_j = \mathbf{a}_j^T \mathbf{x}) \qquad j = 1, \dots, d$$

$$\mathbf{A} = \begin{bmatrix} \boxed{\begin{matrix} a_{11} \\ a_{21} \\ \vdots \\ a_{d1} \end{matrix}} & \begin{matrix} a_{12} \\ a_{22} \\ \vdots \\ a_{d2} \end{matrix} & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \boxed{\begin{matrix} a_{1d'} \\ a_{2d'} \\ \vdots \\ a_{dd'} \end{matrix}} \end{bmatrix}$$

$\mathbf{a}_1 \qquad \mathbf{a}_{d'}$

Linear Dimensionality Reduction

- Unsupervised
  - Principal Component Analysis (PCA) [we will discuss]
  - Independent Component Analysis (ICA)
  - Singular Value Decomposition (SVD)
  - Multi Dimensional Scaling (MDS)
  - Canonical Correlation Analysis (CCA)

- Supervised
  - Linear Discriminant Analysis (LDA) [we will discuss]

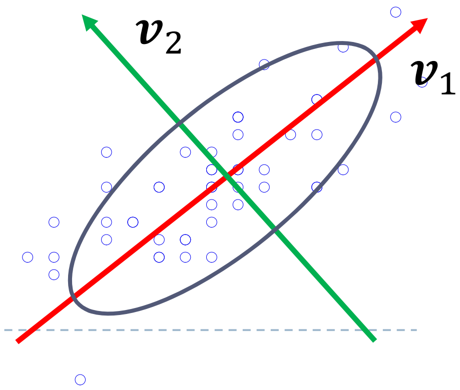
Unsupervised Feature Reduction

- Visualization: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage, communication, or and retrieval.
- Noise removal: to improve accuracy by removing irrelevant features.
  - As a preprocessing step to reduce dimensions for classification tasks

Principal Component Analysis (PCA)

- The "best" subspace:
  - Centered at the sample mean
  - The axes have been rotated to new (principal) axes such that:
    - Principal axis 1 has the highest variance
    - Principal axis 2 has the next highest variance, and so on.
  - The principal axes are uncorrelated
    - Covariance among each pair of the principal axes is zero.

- Goal: reducing the dimensionality of the data while preserving the variation present in the dataset as much as possible.



Principal Component Analysis (PCA)

- Principal Components (PCs): orthogonal vectors that are ordered by the fraction of the total information (variation) in the corresponding directions

- PCs can be found as the "best" eigenvectors of the covariance matrix of the data points.
  - If data has a Gaussian distribution $N(\mu, \Sigma)$, the direction of the largest variance can be found by the eigenvector of Σ that corresponds to the largest eigenvalue of Σ

Covariance Matrix

$$\boldsymbol{\mu}_x = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} = \begin{bmatrix} E(x_1) \\ \vdots \\ E(x_d) \end{bmatrix}$$

$$\boldsymbol{\Sigma} = E\left[(\boldsymbol{x} - \boldsymbol{\mu}_x)(\boldsymbol{x} - \boldsymbol{\mu}_x)^T\right]$$

- ML estimate of covariance matrix from data points $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$:

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\left(\widetilde{\boldsymbol{X}}^T \widetilde{\boldsymbol{X}}\right)$$

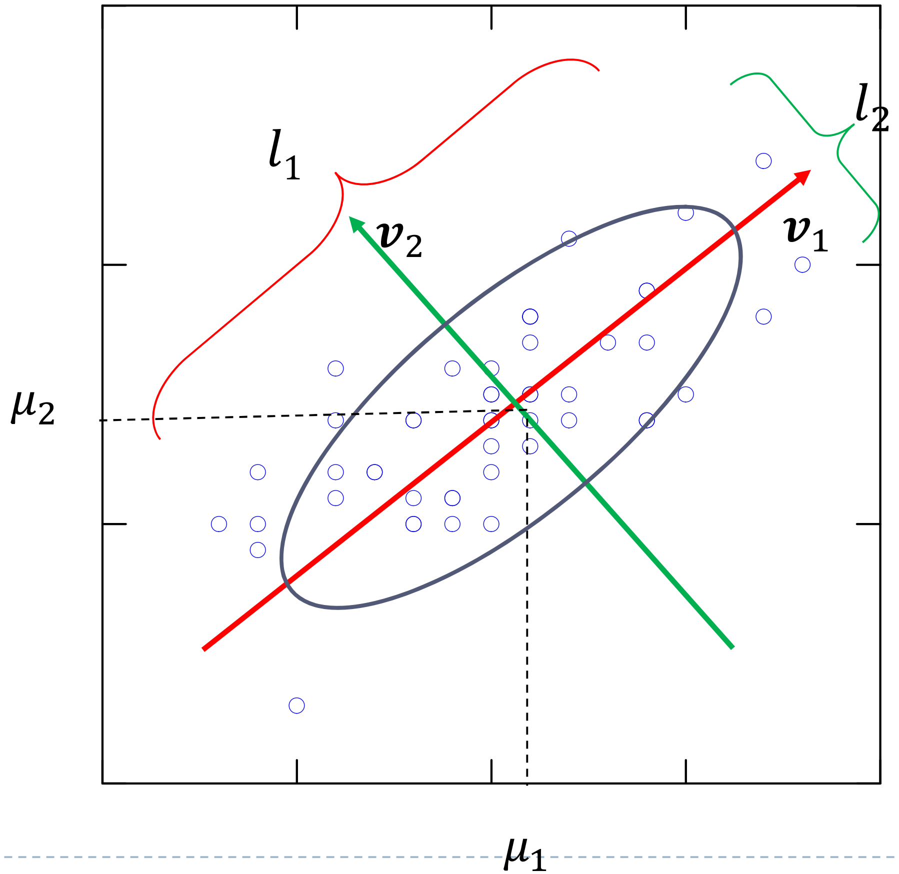
$$\widetilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{x}^{(1)} - \widehat{\boldsymbol{\mu}} \\ \vdots \\ \boldsymbol{x}^{(N)} - \widehat{\boldsymbol{\mu}} \end{bmatrix} \qquad \widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$$

Mean-centered data

Eigenvalues and Eigenvectors: Geometrical Interpretation

- Covariance matrix is a PSD matrix $\boldsymbol{C}$
  - corresponding to a hyper-ellipsoidal in an d-dimensional space

$$\frac{l_1}{l_2} = \sqrt{\frac{\lambda_1}{\lambda_2}}$$



$$\boldsymbol{C} = [\boldsymbol{v}_1\ \boldsymbol{v}_2]\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}[\boldsymbol{v}_1\ \boldsymbol{v}_2]^T$$

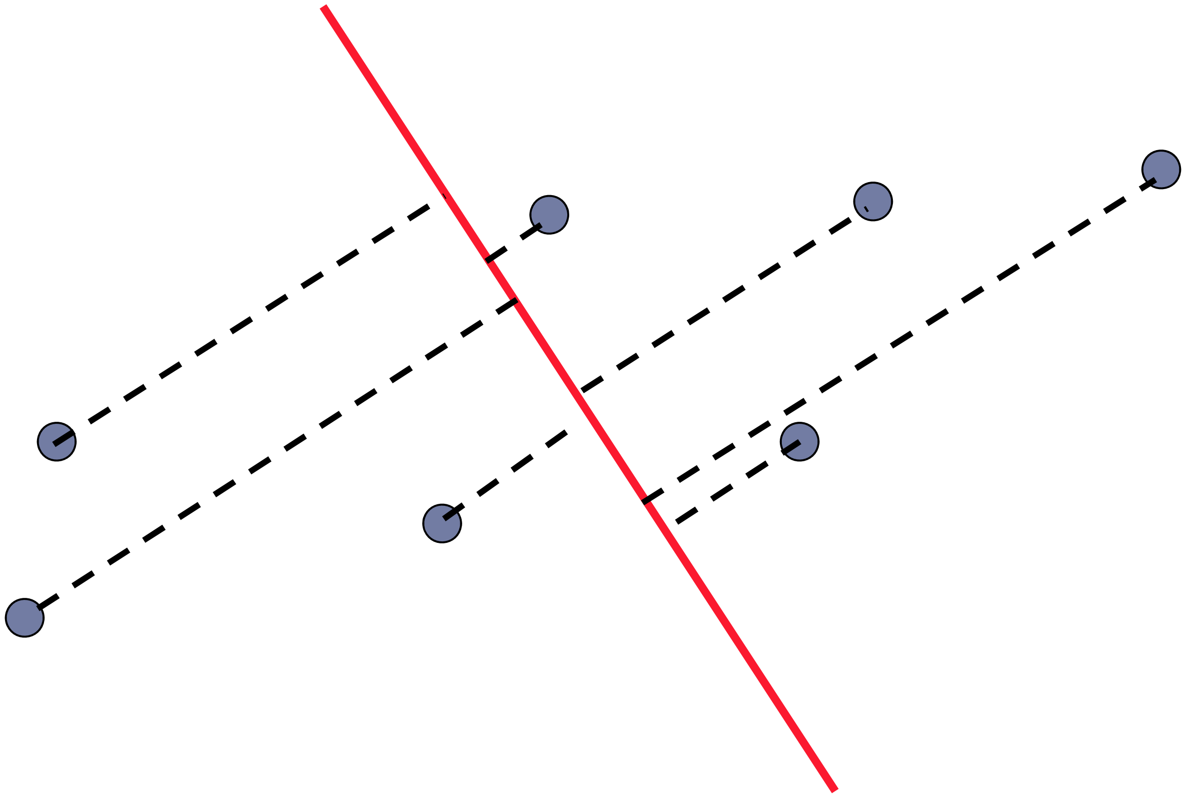
PCA: Steps

- Input: $N \times d$ data matrix $\boldsymbol{X}$ (each row contain a d dimensional data point)
  - $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}^{(i)}$
  - $\widetilde{\boldsymbol{X}} \leftarrow$ Mean value of data points is subtracted from rows of $\boldsymbol{X}$
  - $\boldsymbol{\Sigma} = \frac{1}{N}\widetilde{\boldsymbol{X}}^T\widetilde{\boldsymbol{X}}$ (Covariance matrix)
  - Calculate eigenvalue and eigenvectors of $\boldsymbol{\Sigma}$
  - Pick d' eigenvectors corresponding to the largest eigenvalues and put them in the columns of $\boldsymbol{A} = [\underbrace{\boldsymbol{v}_1}_{\text{First PC}}, \dots, \underbrace{\boldsymbol{v}_{d'}}_{\text{d'-th PC}}]$
  - $\boldsymbol{X}' = \boldsymbol{A}^T\boldsymbol{X}$

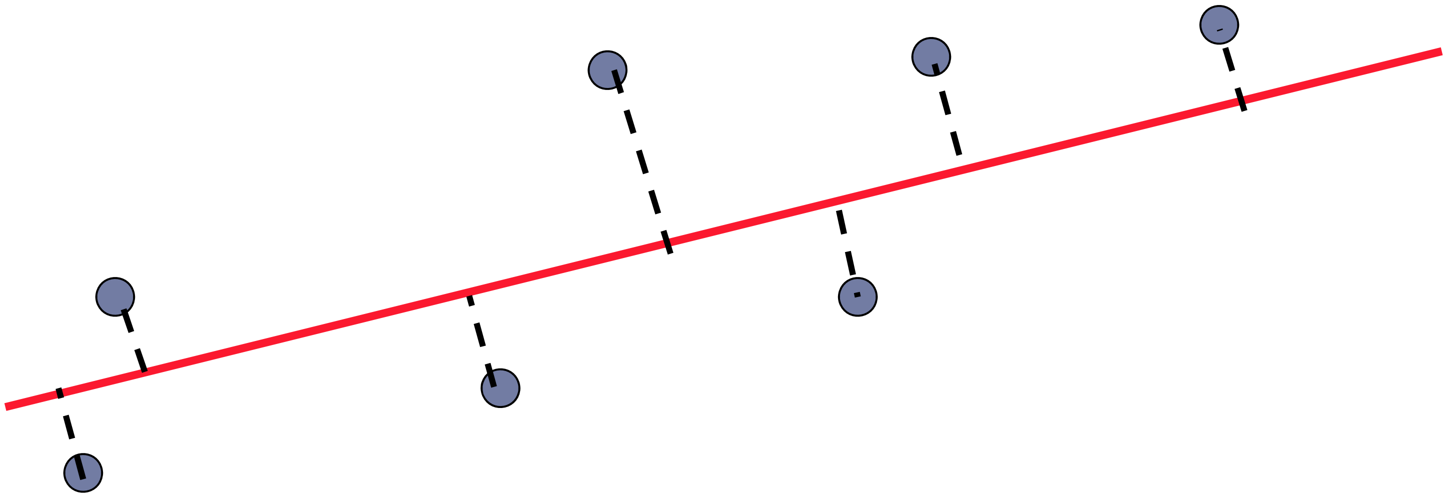
Another Interpretation: Least Squares Error

- PCs are linear least squares fits to samples, each orthogonal to the previous PCs:
  - First PC is a minimum distance fit to a vector in the original feature space
  - Second PC is a minimum distance fit to a vector in the plane perpendicular to the first PC

Another Interpretation: Least Squares Error

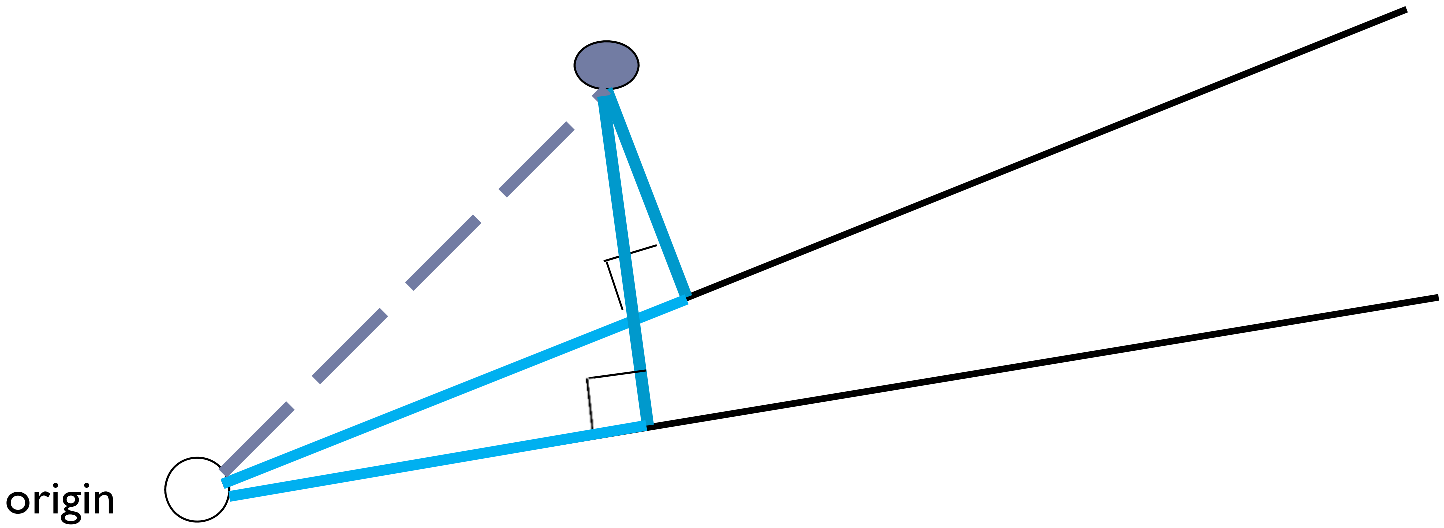


Another Interpretation: Least Squares Error



Least Squares Error and Maximum Variance Views Are Equivalent (1-dim Interpretation)

- Minimizing sum of square distances to the line is equivalent to maximizing the sum of squares of the projections on that line (Pythagoras).



PCA: Uncorrelated Features

$$\boldsymbol{x}' = \boldsymbol{A}^T \boldsymbol{x}$$

$$\boldsymbol{R}_{\boldsymbol{x}'} = E\left[\boldsymbol{x}'\boldsymbol{x}'^T\right] = E[\boldsymbol{A}^T\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{A}] = \boldsymbol{A}^T E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{A} = \boldsymbol{A}^T\boldsymbol{R}_{\boldsymbol{x}}\boldsymbol{A}$$

- If $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_d]$ where $\boldsymbol{a}_1, \dots, \boldsymbol{a}_d$ are orthonormal eighenvectors of $\boldsymbol{R}_{\boldsymbol{x}}$:

$$\boldsymbol{R}_{\boldsymbol{x}'} = \boldsymbol{A}^T\boldsymbol{R}_{\boldsymbol{x}}\boldsymbol{A} = \boldsymbol{A}^T(\boldsymbol{A}^T\boldsymbol{\Lambda}\boldsymbol{A})\boldsymbol{A} = \boldsymbol{\Lambda}$$
$$\Rightarrow \forall i \neq j\ (i, j = 1, \dots, d)\ E\left[\boldsymbol{x}_i'\boldsymbol{x}_j'\right] = 0$$

  then mutually uncorrelated features are obtained

  - Completely uncorrelated features avoid information redundancies

PCA Derivation (Correlation Version): Mean Square Error Approximation

- Incorporating all eigenvectors in $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_d]$:

$$\boldsymbol{x}' = \boldsymbol{A}^T\boldsymbol{x} \Rightarrow \boldsymbol{A}\boldsymbol{x}' = \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{x} = \boldsymbol{x}$$
$$\Rightarrow \boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{x}'$$

- $\Longrightarrow$ If $d' = d$ then $\boldsymbol{x}$ can be reconstructed exactly from $\boldsymbol{x}'$

PCA Derivation (Correlation Version): Mean Square Error Approximation

- Incorporating only d' eigenvectors corresponding to the largest eigenvalues $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_d]$ $(d' < d)$

- It minimizes MSE between $\boldsymbol{x}$ and $\widehat{\boldsymbol{x}} = \boldsymbol{A}^T \boldsymbol{x}'$:

$$J(\boldsymbol{A}) = E[\|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2] = E[\|\boldsymbol{x} - \boldsymbol{A}^T \boldsymbol{x}'\|^2] = E\left[\left\|\sum_{j=d'+1}^{d} x_j' \boldsymbol{a}_j\right\|^2\right]$$

$$= E\left[\sum_{j=d'+1}^{d} \sum_{k=d'+1}^{d} x_j' \boldsymbol{a}_j^T \boldsymbol{a}_k \, x_k'\right] = E\left[\sum_{j=d'+1}^{d} {x_j'}^2\right] = \sum_{j=d'+1}^{d} E\left[{x_j'}^2\right]$$

$$= \sum_{j=d'+1}^{d} \boldsymbol{a}_j^T E[\boldsymbol{x}\boldsymbol{x}^T] \boldsymbol{a}_j == \sum_{j=d'+1}^{d} \boldsymbol{a}_j^T \boldsymbol{R}_x \boldsymbol{a}_j = \sum_{j=d'+1}^{d} \lambda_j$$

Sum of the $d - d'$ smallest eigenvalues

PCA Derivation (Correlation Version): Relation between Eigenvalues and Variances

- The j-th largest eigenvalue of $\boldsymbol{R_x}$ is the variance on the j-th PC:

$$var(x_j') = \boldsymbol{a}_j^T \boldsymbol{R_x} \boldsymbol{a}_j = \lambda_j$$

PCA Derivation (Correlation Version): Mean Square Error Approximation

- In general, it can also be shown MSE is minimized compared to any other approximation of $\boldsymbol{x}$ by any d'-dimensional orthonormal basis
  - without first assuming that the axes are eigenvectors of the correlation matrix, this result can also be obtained.

- If the data is mean-centered in advance, $\boldsymbol{R_x}$ and $\boldsymbol{C_x}$ (covariance matrix) will be the same.
  - However, in the correlation version when $\boldsymbol{C_x} \neq \boldsymbol{R_x}$ the approximation is not, in general, a good one (although it is a minimum MSE solution)

PCA on Faces: "Eigenfaces"

- ORL Database

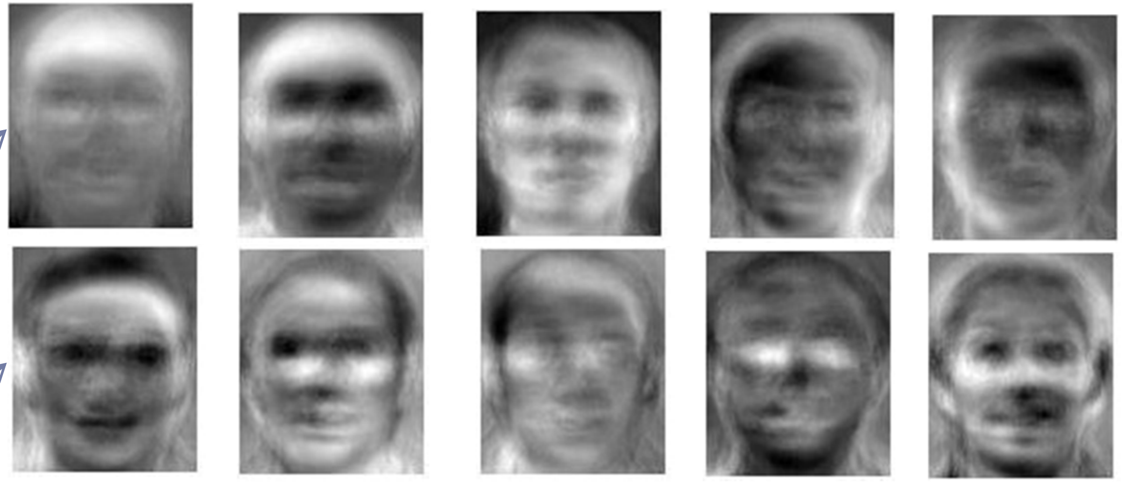
Some Images

PCA on Faces: “Eigenfaces”

Average face

1st PC

6th PC



For eigen faces
“gray” = 0,
“white” > 0,
“black” < 0

PCA on Faces:

$\boldsymbol{x}$ is a $112 \times 92 = 10304$ dimensional vector containing intensity of the pixels of this image

Feature vector=$[x'_1, x'_2, \dots, x'_{d'}]$

$x'_i = PC_i^T \boldsymbol{x} \longrightarrow$ The projection of $\boldsymbol{x}$ on the i-th PC

$=$

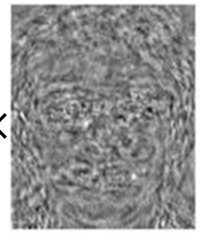
Average Face

$+x'_1 \times$

$+x'_2 \times$

$+ \cdots$

$+x'_{256} \times$



PCA on Faces: Reconstructed Face

d'=1

d'=2

d'=4

d'=8

d'=16

d'=32

d'=64

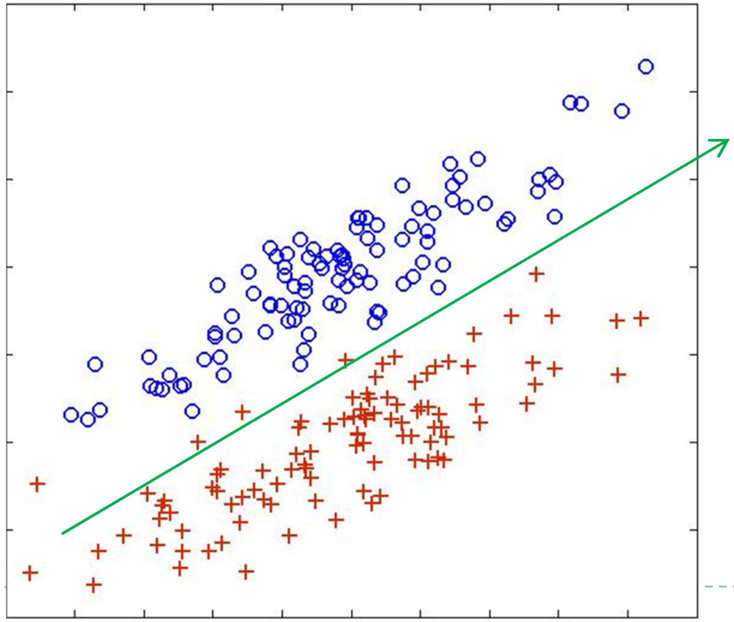
d'=128

d'=256

Original Image

PCA Drawback

- An excellent information packing transform does not necessarily lead to a good class separability.
  - The directions of the maximum variance may be useless for classification purpose



Independent Component Analysis (ICA)

- PCA:
  - The transformed dimensions will be uncorrelated from each other
  - Orthogonal linear transform
  - Only uses second order statistics (i.e., covariance matrix)

- ICA:
  - The transformed dimensions will be as independent as possible.
  - Non-orthogonal linear transform
  - High-order statistics are used

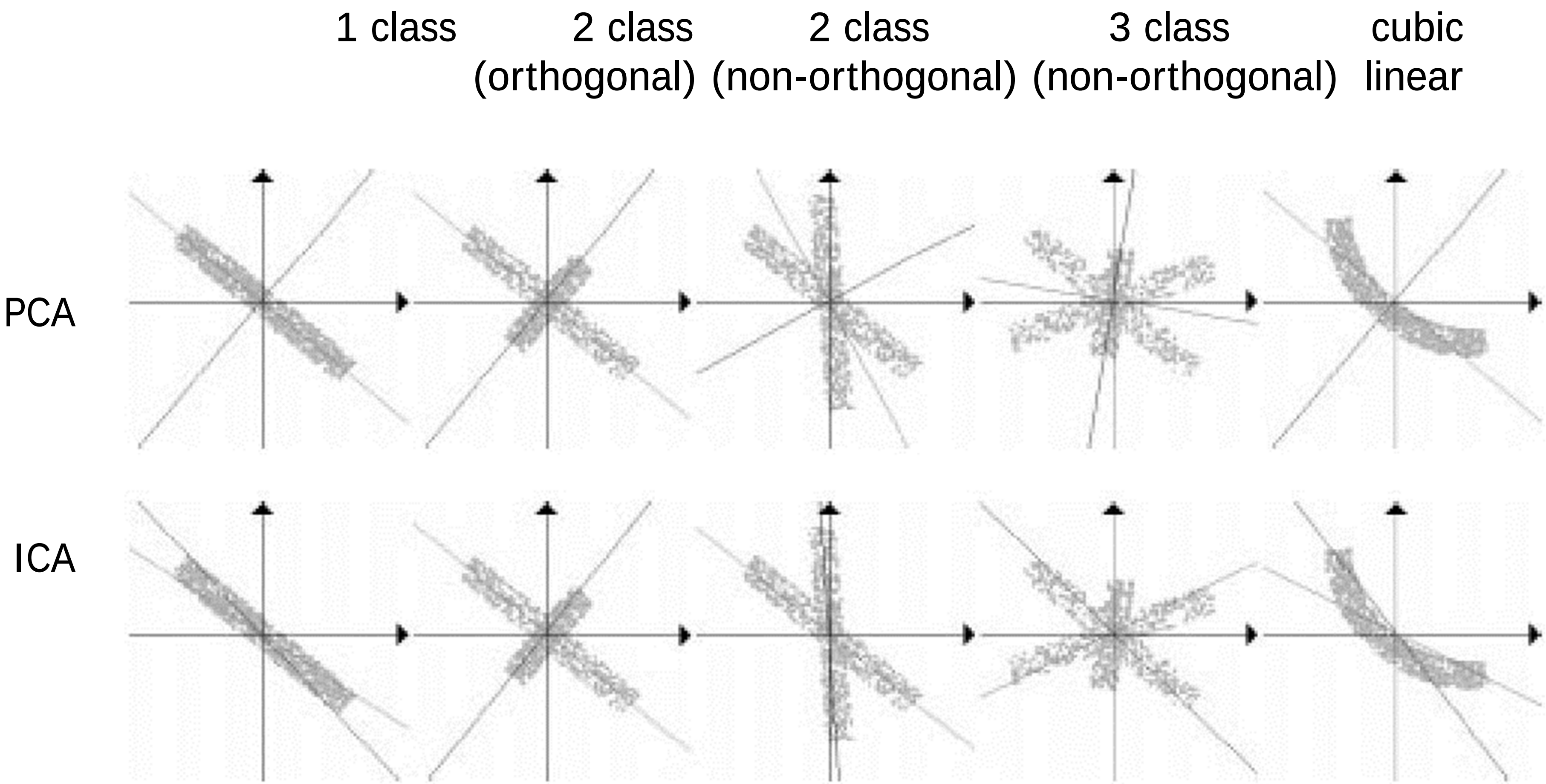
Uncorrelated and Independent

Uncorrelated: $cov(X_1, X_2) = 0$
Independent: $P(X_1, X_2) = P(X_1)P(X_2)$

- Gaussian
  - Independent $\Leftrightarrow$ Uncorrelated

- Non-Gaussian
  - Independent $\Rightarrow$ Uncorrelated
  - Uncorrelated $\nRightarrow$ Independent

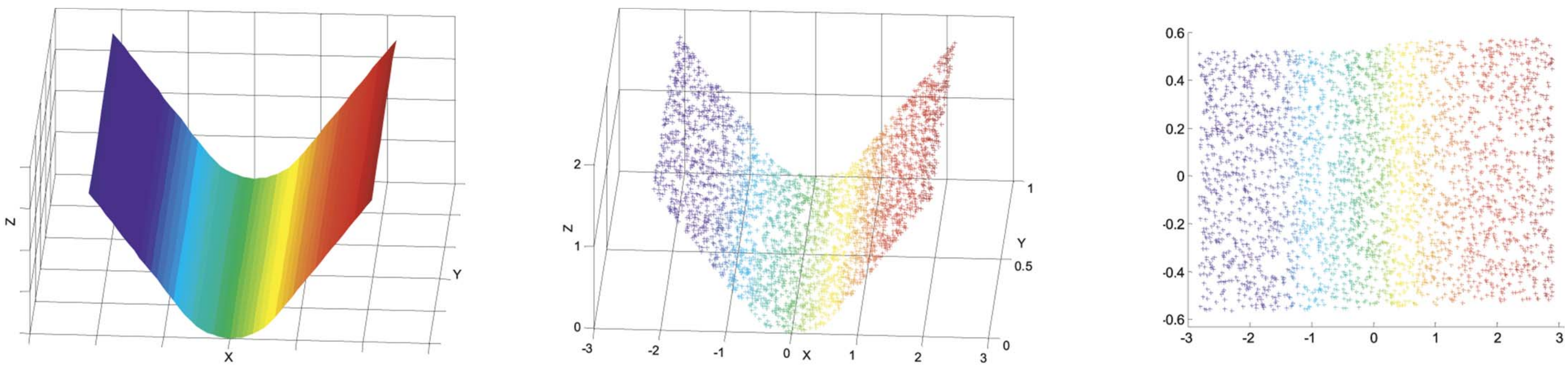
PCA vs. ICA



1 class
2 class (orthogonal)
2 class (non-orthogonal)
3 class (non-orthogonal)
cubic linear
PCA
ICA

Kernel PCA

- Kernel extension of PCA



data (approximately) lies on
a lower dimensional non-linear space

Kernel PCA

- Hilbert space: $\boldsymbol{x} \rightarrow \boldsymbol{\phi}(\boldsymbol{x})$ (Nonlinear extension of PCA)

$$\boldsymbol{C} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)\boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)^{T}$$

- All eigenvectors of $\boldsymbol{C}$ lie in the span of the mapped data points,

$$\boldsymbol{C}\boldsymbol{v} = \lambda\boldsymbol{v}$$

$$\boldsymbol{v} = \sum_{i=1}^{N} \boldsymbol{\phi}\left(\boldsymbol{x}^{(i)}\right)$$

- After some algebra, we have:

$$\boldsymbol{K}\boldsymbol{\alpha} = N\lambda\boldsymbol{\alpha}$$

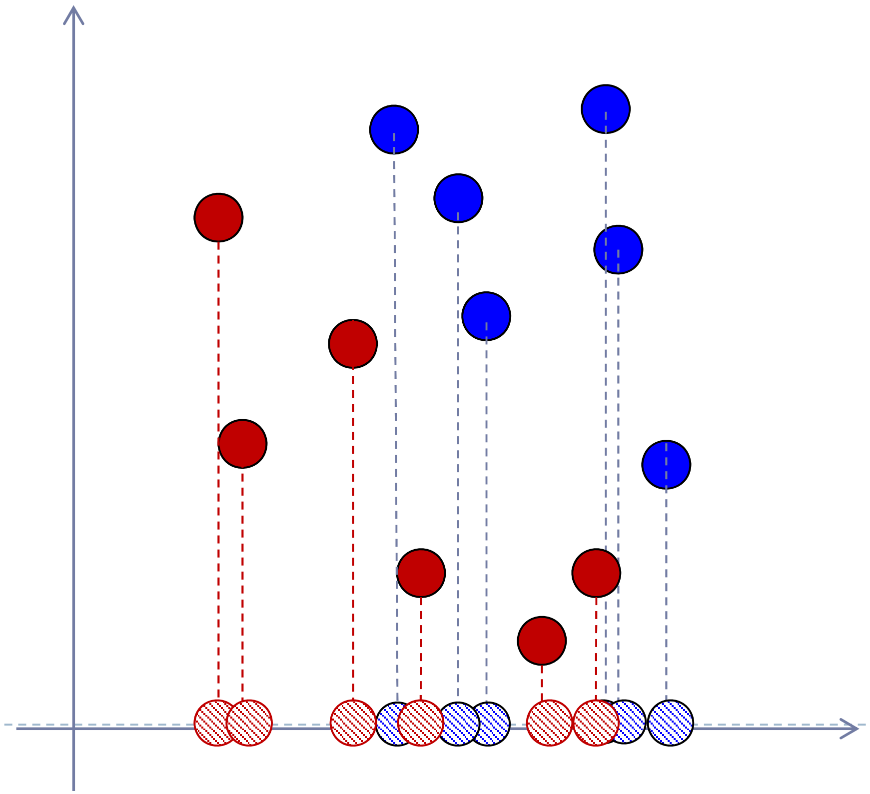
Linear Discriminant Analysis (LDA)

- Supervised feature extraction

- Fisher's Linear Discriminant Analysis :
  - Dimensionality reduction
    - Finds linear combinations of features with large ratios of between-groups to within-groups scatters (as discriminant new variables)
  - Classification
    - Example: Predicts the class of an observation x by the class whose mean vector is the closest to x in the space of the discriminant variables

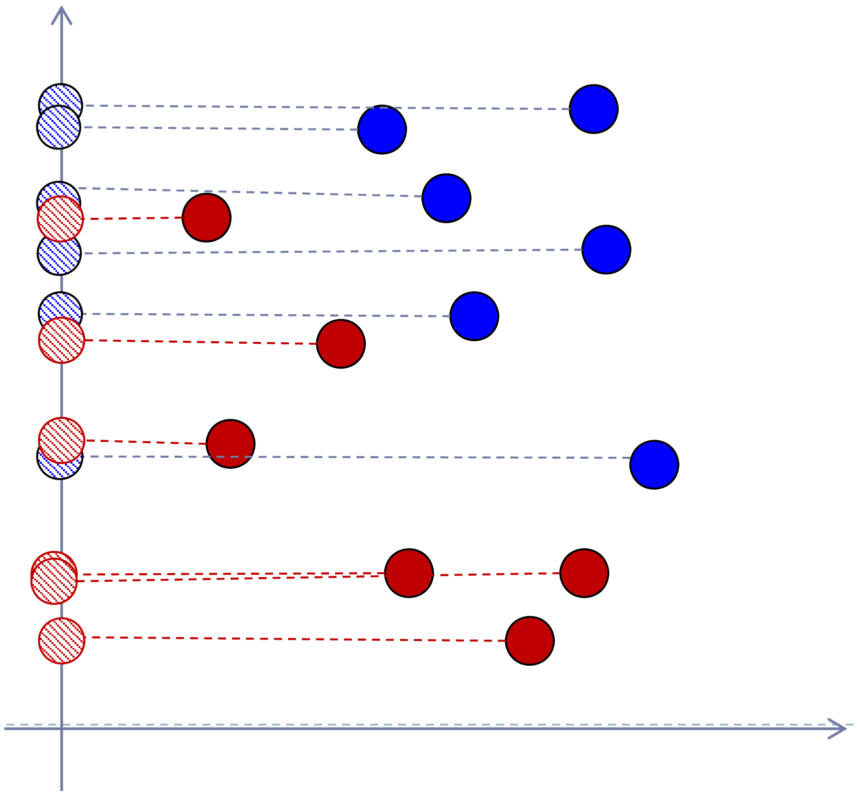
Good Projection for Classification

- What is a good criterion?
  - Separating different classes in the projected space
  - As opposed to PCA, we use also the labels of the training data



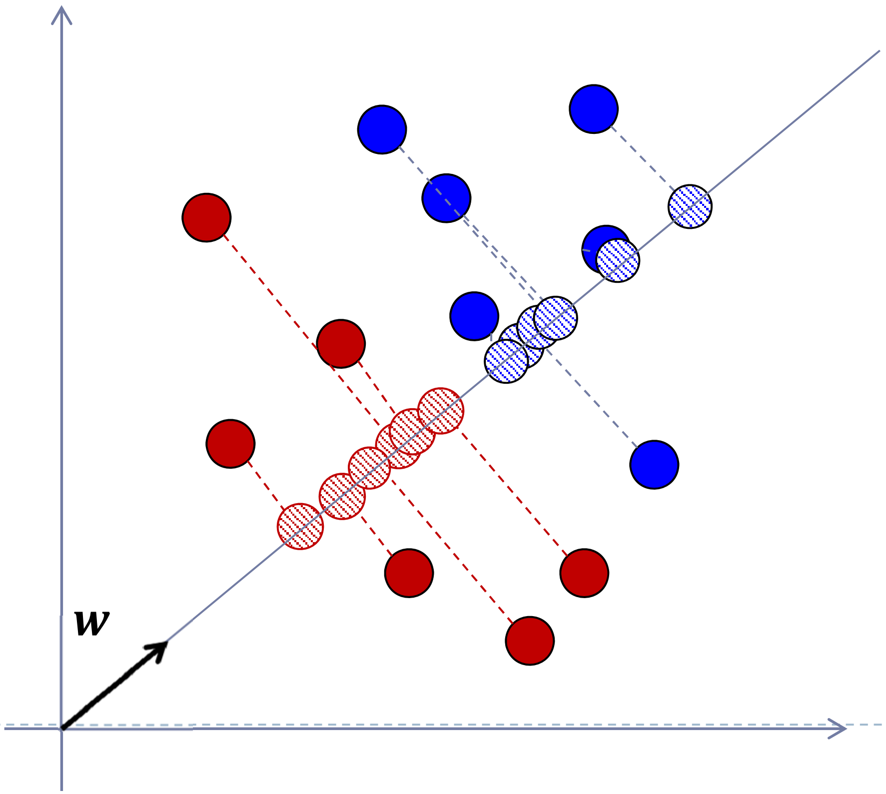
Good Projection for Classification

- What is a good criterion?
  - Separating different classes in the projected space
  - As opposed to PCA, we use also the labels of the training data



Good Projection for Classification

- What is a good criterion?
  - Separating different classes in the projected space
  - As opposed to PCA, we use also the labels of the training data



LDA Problem

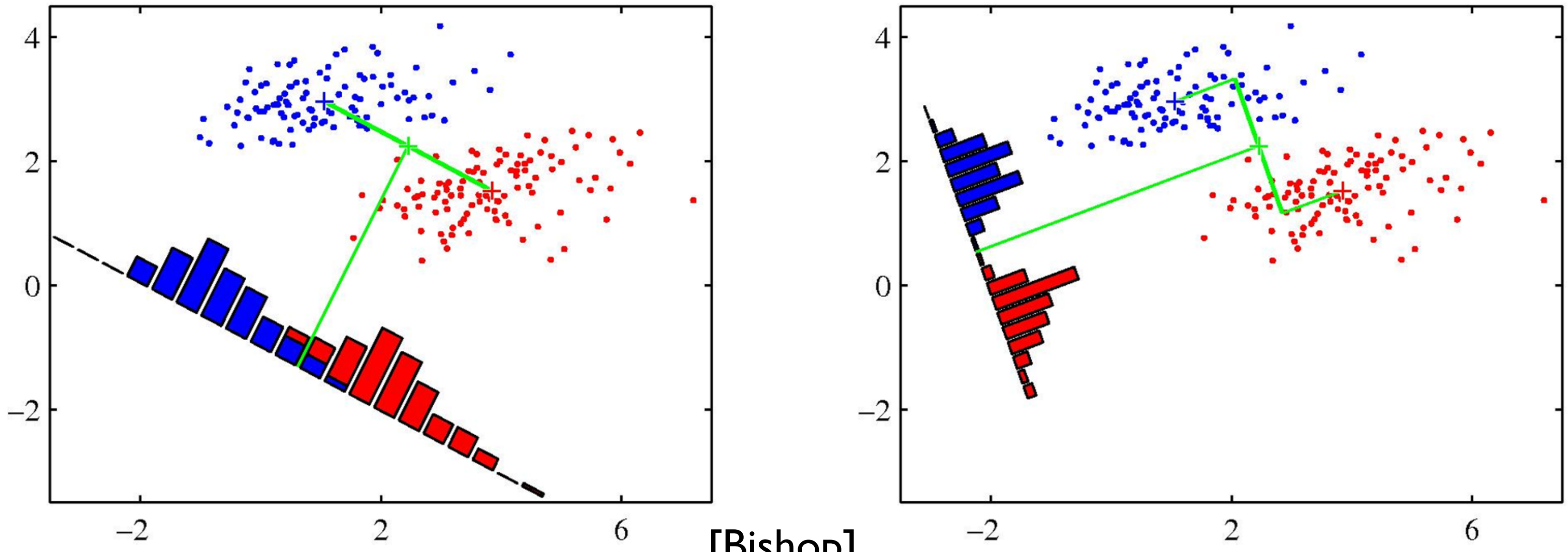
- Problem definition:
  - $C = 2$ classes
  - $\left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ training samples with N_1 samples from the first class ($\mathcal{C}_1$) and N_2 samples from the second class ($\mathcal{C}_2$)
  - Goal: finding the best direction $\boldsymbol{w}$ that we hope will enable accurate classification

- The projection of sample $\boldsymbol{x}$ onto a line in direction $\boldsymbol{w}$ is $\boldsymbol{w}^T\boldsymbol{x}$

- What is the measure of the separation between the projected points of different classes?

Measure of Separation in the Projected Direction

- Is the direction of the line jointing the class means is a good candidate for $\boldsymbol{w}$?



[Bishop]

$$\boldsymbol{\mu}_1 = \frac{\sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_1} \boldsymbol{x}^{(i)}}{N_1} \qquad \boldsymbol{\mu}_2 = \frac{\sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_2} \boldsymbol{x}^{(i)}}{N_2}$$

Measure of Separation in the Projected Direction

- The direction of the line jointing the class means is the solution of the following problem:
  - Maximizes the separation of the projected class means

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = (\mu_1' - \mu_2')^2$$
$$\text{s.t.}\ \|\boldsymbol{w}\| = 1$$

$$\mu_1' = \boldsymbol{w}^T\ \boldsymbol{\mu}_1$$
$$\mu_2' = \boldsymbol{w}^T\ \boldsymbol{\mu}_2$$

- What is the problem with the criteria considering only $|\mu_1' - \mu_2'|$?
  - It does not consider the variances of the classes

LDA Criteria

- Fisher idea: maximize a function that will give
  - large separation between the projected class means
  - while also achieving a small variance within each class, thereby minimizing the class overlap.

$$J(\boldsymbol{w}) = \frac{|\mu_1' - \mu_2'|^2}{s_1'^2 + s_2'^2}$$

LDA Criteria

- The scatters of the original data are:

$$s_1^2 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_1} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_1 \right\|^2$$

$$s_2^2 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_2} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{\mu}_2 \right\|^2$$

- The scatters of projected data are:

$$s_1'^2 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_1} \left\| \boldsymbol{w}^T \boldsymbol{x}^{(i)} - \boldsymbol{w}^T \boldsymbol{\mu}_1 \right\|^2$$

$$s_2'^2 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_2} \left\| \boldsymbol{w}^T \boldsymbol{x}^{(i)} - \boldsymbol{w}^T \boldsymbol{\mu}_2 \right\|^2$$

LDA Criteria

$$J(\boldsymbol{w}) = \frac{|\mu_1' - \mu_2'|^2}{s_1'^2 + s_2'^2}$$

$$|\mu_1' - \mu_2'|^2 = |\boldsymbol{w}^T \boldsymbol{\mu}_1 - \boldsymbol{w}^T \boldsymbol{\mu}_2|^2$$

$$= \boldsymbol{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \boldsymbol{w}$$

$$s_1'^2 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_1} \left\| \boldsymbol{w}^T \boldsymbol{x}^{(i)} - \boldsymbol{w}^T \boldsymbol{\mu}_1 \right\|^2$$

$$= \boldsymbol{w}^T \left(\sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_1\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_1\right)^T \right) \boldsymbol{w}$$

LDA Criteria

$$J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S}_W \boldsymbol{w}}$$

Between-class scatter matrix ⟵ $\boldsymbol{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$

Within-class scatter matrix ⟵ $\boldsymbol{S}_W = \boldsymbol{S}_1 + \boldsymbol{S}_2$

$$\boldsymbol{S}_1 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_1\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_1\right)^T$$

$$\boldsymbol{S}_2 = \sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_2} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_2\right)\left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_2\right)^T$$

scatter matrix=N×covariance matrix

LDA Derivation

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{\frac{\partial \mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\partial \mathbf{w}} \times \mathbf{w}^T \mathbf{S}_W \mathbf{w} - \frac{\partial \mathbf{w}^T \mathbf{S}_W \mathbf{w}}{\partial \mathbf{w}} \times \mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\left(\mathbf{w}^T \mathbf{S}_W \mathbf{w}\right)^2} = \frac{\left(2\mathbf{S}_B \mathbf{w}\right)\mathbf{w}^T \mathbf{S}_W \mathbf{w} - \left(2\mathbf{S}_W \mathbf{w}\right)\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\left(\mathbf{w}^T \mathbf{S}_W \mathbf{w}\right)^2}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w}$$

LDA Derivation

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w} \xrightarrow{\mathbf{S}_\mathrm{W}\ \text{full-rank}} \mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}$$

- $\boldsymbol{S}_B \boldsymbol{x}$ for any vector $\boldsymbol{x}$ points in the same direction as $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$:

$$\mathbf{S}_B \mathbf{x} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \mathbf{x} = \alpha(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

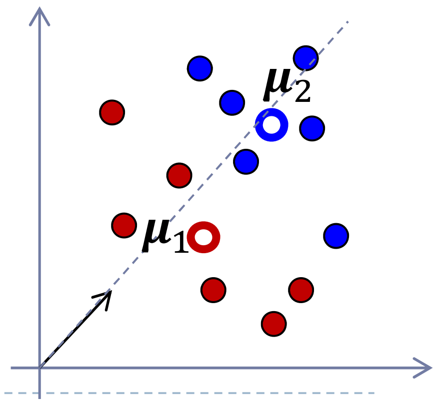
$$\mathbf{w} = \mathbf{S}_W^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

- Thus, we can solve the eigenvalue problem immediately

LDA Algorithm

- $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ ← mean of samples of class 1 and 2 respectively
- $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ ← scatter matrix of class 1 and 2 respectively
- $\boldsymbol{S}_W = \boldsymbol{S}_1 + \boldsymbol{S}_2$
- $\boldsymbol{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$

- Feature Extraction
  - $\boldsymbol{w} = \boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ as the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$
- Classification
  - $\boldsymbol{w} = \boldsymbol{S}_w^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$
  - Using a threshold on $\boldsymbol{w}^T\boldsymbol{x}$, we can classify $\boldsymbol{x}$



Multi-Class LDA (MDA)

- $C > 2$: the natural generalization of LDA involves $C - 1$ discriminant functions.
  - The projection from a d-dimensional space to a $(C - 1)$-dimensional space (tacitly assumed that $d \geq C$).

$$\boldsymbol{S}_W = \sum_{j=1}^{C} \boldsymbol{S}_j$$

$$\boldsymbol{S}_B = \sum\nolimits_{j=1}^{C} N_j (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$$

$$\boldsymbol{\mu}_j = \frac{\sum_{\boldsymbol{x}^{(i)} \in \mathcal{C}_j} \boldsymbol{x}^{(i)}}{N_j} \quad j = 1, \dots, C$$

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{N} \boldsymbol{x}^{(i)}}{N}$$

$$\boldsymbol{S}_j = \sum\nolimits_{\boldsymbol{x}^{(i)} \in \mathcal{C}_j} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T \quad j = 1, \dots, C$$

Multi-Class LDA

- $\boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \ \dots \boldsymbol{w}_{C-1}]$
- $\boldsymbol{x}' = \boldsymbol{W}^T\boldsymbol{x}$

- Means and scatters after transform $\boldsymbol{x}' = \boldsymbol{W}^T\boldsymbol{x}$:
  - $\boldsymbol{S}'_B = \boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W}$
  - $\boldsymbol{S}'_W = \boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W}$

Multi-Class LDA: Objective Function

- We seek a transformation matrix $\boldsymbol{W}$ that in some sense "maximizes the ratio of the between-class scatter to the within-class scatter".

- A simple scalar measure of scatter is the **determinant of the scatter matrix**.

Multi-Class LDA: Objective Function

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

determinant

- The solution of the problem where $\boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \ \dots \boldsymbol{w}_{C-1}]$:

$$\boldsymbol{S}_B \boldsymbol{w}_i = \lambda_i \boldsymbol{S}_W \boldsymbol{w}_i$$

- It is a generalized eigenvectors problem.

Multi-Class LDA: $d' \leq C - 1$

- $rank(\boldsymbol{S}_B) \leq C - 1$
  - $\boldsymbol{S}_B$ is the sum of C matrices $(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T$ of rank (at most) one and only $C - 1$ of these are independent,
  - $\Rightarrow$ atmost $C - 1$ nonzero eigenvalues and the desired weight vectors correspond to these nonzero eigenvalues.

Multi-Class LDA: Other Objective Functions

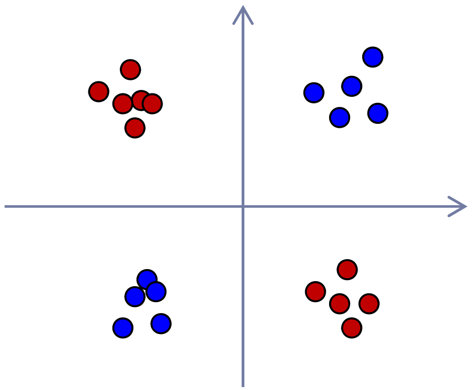
- There are many possible choices of criterion for multi-class LDA, e.g.:

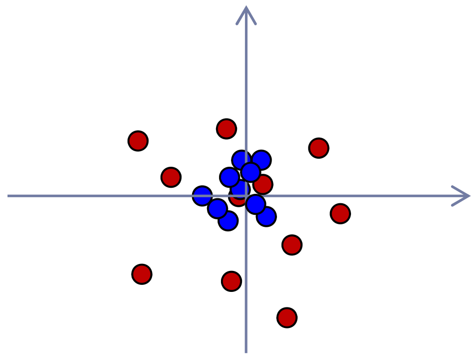
$$J(\boldsymbol{W}) = tr(\boldsymbol{S}'^{-1}_{W}\boldsymbol{S}'_{B}) = tr\left((\boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W})^{-1}(\boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W})\right)$$

- The solution is given by solving a generalized eigenvalue problem $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$
  - Solution: eigen vectors corresponding to the largest eigen values constitute the new variables

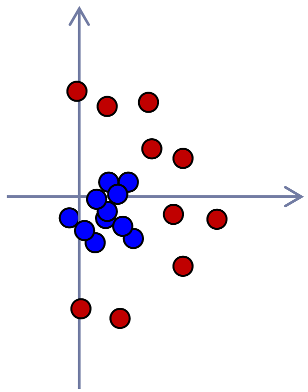
LDA Criterion Limitation

- When $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$, LDA criterion can not lead to a proper projection ($J(\boldsymbol{w}) = 0$)
  - However, discriminatory information in the scatter of the data may be helpful





- If classes are non-linearly separable they may have large overlap when projected to any line
- LDA implicitly assumes Gaussian distribution of samples of each class



Issues in LDA

- Singularity or undersampled problem (when $N < d$)
  - Example: gene expression data, images, text documents

- Can reduces dimension only to $d' \leq C - 1$ (unlike PCA)

- Approaches to avoid these problems:
  - PCA+LDA, Regularized LDA, Locally FDA (LFDA), etc.

Summary

- Although LDA often provide more suitable features for classification tasks, PCA might outperform LDA in some situations such as:
  - when the number of samples per class is small (overfitting problem of LDA)
  - when the training data non-uniformly sample the underlying distribution
  - when the number of the desired features is more than $C - 1$

- Advances in the recent decade:
  - Semi-supervised feature extraction
  - Nonlinear dimensionality reduction