PCA and LDA

Man-Wai MAK

Dept. of Electronic and Information Engineering,
The Hong Kong Polytechnic University

enmwmak@polyu.edu.hk
http://www.eie.polyu.edu.hk/∼mwmak

References:

S.J.D. Prince, *Computer Vision: Models Learning and Inference*, Cambridge University Press, 2012

C. Bishop, "*Pattern Recognition and Machine Learning*", Appendix E, Springer, 2006.

October 24, 2019

Overview

1. Dimension Reduction
   - Why Dimension Reduction
   - Dimension Reduction: Reduce to 1-Dim
2. Principle Component Analysis
   - Derivation of PCA
   - PCA on High-Dimensional Data
   - Eigenface
3. Linear Discriminant Analysis
   - LDA on 2-Class Problems
   - LDA on Multi-class Problems

Why Dimension Reduction

- Many applications produce high-dimensional vectors
  - In face recognition, if an image has size 360×260 pixels, the dimension is 93600.
  - In hand-writing digit recognition, if a digit occupies 28×28 pixels, the dimension is 784.
  - In speaker recognition, the dim can be as high as 61440 per utterance.
- High-dim feature vectors can easily cause the curse-of-dimensionality problem.
- **Redundancy**: Some of the elements in the feature vectors are strongly correlated, meaning that knowing one element will also know some other elements.
- **Irrelevancy**: Some elements in the feature vectors are irrelevant to the classification task.

Dimension Reduction

- Given a feature vector $\mathbf{x} \in \mathbb{R}^D$, dimensionality reduction aims to find a low dimensional representation $\mathbf{h} \in \mathbb{R}^M$ that can approximately explain $\mathbf{x}$:

$$\mathbf{x} \approx f(\mathbf{h}, \boldsymbol{\theta}) \tag{1}$$

  where $f(\cdot, \cdot)$ is a function that takes the hidden variable $\mathbf{h}$ and a set of parameters $\boldsymbol{\theta}$ and $M \leq D$.

- Typically, we choose the function family $f(\cdot, \cdot)$ and then learn $\mathbf{h}$ and $\boldsymbol{\theta}$ from training data.

- **Least squares criterion**: Given N training vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^D$, we find the parameters $\boldsymbol{\theta}$ and latent variables $\mathbf{h}_i$'s that minimize the sum of squared error:

$$\hat{\boldsymbol{\theta}}, \{\hat{\mathbf{h}}_i\}_{i=1}^N = \underset{\boldsymbol{\theta}, \{\mathbf{h}_i\}_{i=1}^N}{\operatorname{argmin}} \left\{ \sum_{i=1}^N [\mathbf{x}_i - f(\mathbf{h}_i, \boldsymbol{\theta})]^\mathsf{T} [\mathbf{x}_i - f(\mathbf{h}_i, \boldsymbol{\theta})] \right\} \tag{2}$$

Dimension Reduction: Reduce to 1-Dim

- Approximate vector $\mathbf{x}_i$ by a scalar value h_i plus the global mean $\boldsymbol{\mu}$:

$$\mathbf{x}_i \approx \boldsymbol{\phi} h_i + \boldsymbol{\mu}, \text{ where } \boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i, \qquad \boldsymbol{\phi} \in \mathbb{R}^{D\times 1}$$

- Assuming $\boldsymbol{\mu} = \mathbf{0}$ or vectors have been mean-subtracted, i.e., $\mathbf{x}_i \leftarrow \mathbf{x}_i - \boldsymbol{\mu}\ \forall i$, we have

$$\mathbf{x}_i \approx \boldsymbol{\phi} h_i$$

- The least squares criterion becomes:

$$\begin{aligned} \hat{\boldsymbol{\phi}}, \{\hat{h}_i\}_{i=1}^N &= \underset{\boldsymbol{\phi}, \{h_i\}_{i=1}^N}{\operatorname{argmin}}\ E(\boldsymbol{\phi}, \{h_i\}) \\ &= \underset{\boldsymbol{\phi}, \{h_i\}_{i=1}^N}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} [\mathbf{x}_i - \boldsymbol{\phi} h_i]^{\mathsf{T}} [\mathbf{x}_i - \boldsymbol{\phi} h_i] \right\} \end{aligned} \tag{3}$$

Dimension Reduction: Reduce to 1-Dim

- Eq. 3 has a problem in that it does not have a unique solution. If we multiply ϕ by any constant α and divide h_i's by the same constant we get the same cost, i.e., $\alpha\boldsymbol{\phi} \cdot \frac{h_i}{\alpha} = \boldsymbol{\phi} h_i$.
- We make the solution unique by constraining $\|\phi\|^2 = 1$ using a Lagrange multiplier:

$$
\begin{aligned}
L(\boldsymbol{\phi}, \{h_i\}) &= E(\boldsymbol{\phi}, \{h_i\}) + \lambda(\boldsymbol{\phi}^\mathsf{T}\boldsymbol{\phi} - 1) \\
&= \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\phi} h_i)^\mathsf{T}(\mathbf{x}_i - \boldsymbol{\phi} h_i) + \lambda(\boldsymbol{\phi}^\mathsf{T}\boldsymbol{\phi} - 1) \\
&= \sum_{i=1}^{N} \mathbf{x}^\mathsf{T}\mathbf{x}_i - 2h_i\boldsymbol{\phi}^\mathsf{T}\mathbf{x}_i + h_i^2 + \lambda(\boldsymbol{\phi}^\mathsf{T}\boldsymbol{\phi} - 1)
\end{aligned}
$$

Dimension Reduction: Reduce to 1-Dim

- Setting $\frac{\partial L}{\partial \boldsymbol{\phi}} = \mathbf{0}$ and $\frac{\partial L}{\partial h_i} = 0$, we obtain:

$$\sum_i \mathbf{x}_i \hat{h}_i = \lambda \hat{\boldsymbol{\phi}} \quad \text{and} \quad \hat{\boldsymbol{\phi}}^{\mathsf{T}} \mathbf{x}_i = \hat{h}_i = \mathbf{x}_i^{\mathsf{T}} \hat{\boldsymbol{\phi}}$$

- Hence,

$$\begin{aligned} \sum_i \mathbf{x}_i \left(\mathbf{x}_i^{\mathsf{T}} \hat{\boldsymbol{\phi}} \right) = \left(\sum_i \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \right) \hat{\boldsymbol{\phi}} &= \lambda \hat{\boldsymbol{\phi}} \\ \Longrightarrow \mathbf{S} \hat{\boldsymbol{\phi}} &= \lambda \hat{\boldsymbol{\phi}} \end{aligned}$$

  where $\mathbf{S}$ is the covariance matrix of training data.[1]

- Therefore, $\hat{\boldsymbol{\phi}}$ is the first eigenvector of $\mathbf{S}$.

[1]Note that $\mathbf{x}_i$'s have been mean subtracted.

Dimension Reduction: Reduce to 1-Dim

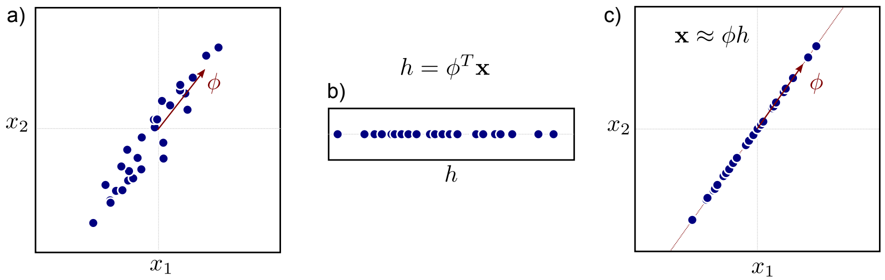
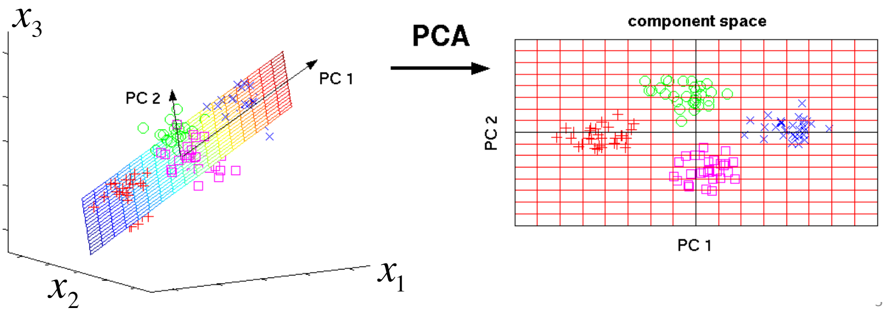


Figure 13.19 Reduction to a single dimension. a) Original data and direction $\boldsymbol{\phi}$ of maximum variance. b) The data are projected onto $\boldsymbol{\phi}$ to produce a one dimensional representation. c) To reconstruct the data, we re-multiply by $\boldsymbol{\phi}$. Most of the original variation is retained. PCA extends this model to project high dimensional data onto the K orthogonal dimensions with the most variance, to produce a K dimensional representation.

Dimension Reduction: 3D to 2D



- In PCA, the hidden variables $\{\mathbf{h}_i\}$ are multi-dimensional and ϕ becomes a rectangular matrix $\mathbf{\Phi} = [\phi_1 \; \phi_2 \; \cdots \; \phi_M]$, where $M \leq D$.
- Each components of $\mathbf{h}_i$ weights one column of matrix $\mathbf{\Phi}$ so that data is approximated as

$$\mathbf{x}_i \approx \mathbf{\Phi}\mathbf{h}_i, \qquad i = 1, \ldots, N$$

- The cost function is[2]

$$\begin{aligned} \hat{\mathbf{\Phi}}, \{\hat{\mathbf{h}}_i\}_{i=1}^N &= \underset{\mathbf{\Phi}, \{\mathbf{h}_i\}_{i=1}^N}{\operatorname{argmin}} \; E\left(\mathbf{\Phi}, \{\mathbf{h}_i\}_{i=1}^N\right) \\ &= \underset{\mathbf{\Phi}, \{\mathbf{h}_i\}_{i=1}^N}{\operatorname{argmin}} \left\{ \sum_{i=1}^N [\mathbf{x}_i - \mathbf{\Phi}\mathbf{h}_i]^\mathsf{T} [\mathbf{x}_i - \mathbf{\Phi}\mathbf{h}_i] \right\} \end{aligned} \tag{4}$$

[2] Note that we have defined $\boldsymbol{\theta} \equiv \mathbf{\Phi}$ in Eq. 2.

- To solve the non-uniqueness problem in Eq. 4, we enforce $\phi_d^{\mathsf{T}}\phi_d = 1$, $d = 1, \ldots, M$, using a set of Lagrange multipliers $\{\lambda_d\}_{d=1}^{M}$:

$$\begin{aligned} L(\mathbf{\Phi}, \{\mathbf{h}_i\}) &= \sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{\Phi}\mathbf{h}_i)^{\mathsf{T}}(\mathbf{x}_i - \mathbf{\Phi}\mathbf{h}_i) + \sum_{d=1}^{M}\lambda_d(\phi_d^{\mathsf{T}}\phi_d - 1) \\ &= \sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{\Phi}\mathbf{h}_i)^{\mathsf{T}}(\mathbf{x}_i - \mathbf{\Phi}\mathbf{h}_i) + \mathrm{tr}\{\mathbf{\Phi}\mathbf{\Lambda}_M\mathbf{\Phi}^{\mathsf{T}} - \mathbf{\Lambda}\} \\ &= \sum_{i=1}^{N}\mathbf{x}^{\mathsf{T}}\mathbf{x}_i - 2\mathbf{h}_i^{\mathsf{T}}\mathbf{\Phi}^{\mathsf{T}}\mathbf{x}_i + \mathbf{h}_i^{\mathsf{T}}\mathbf{h}_i + \mathrm{tr}\{\mathbf{\Phi}\mathbf{\Lambda}_M\mathbf{\Phi}^{\mathsf{T}} - \mathbf{\Lambda}\} \end{aligned} \tag{5}$$

where $\mathbf{h}_i \in \mathbb{R}^M$, $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_M, 0, \ldots, 0\} \in \mathbb{R}^{D\times D}$,
$\mathbf{\Lambda}_M = \mathrm{diag}\{\lambda_1, \ldots, \lambda_M\} \in \mathbb{R}^{M\times M}$, and
$\mathbf{\Phi} = [\phi_1 \ \phi_2 \ \cdots \ \phi_M] \in \mathbb{R}^{D\times M}$.

- Setting $\frac{\partial L}{\partial \mathbf{\Phi}} = \mathbf{0}$ and $\frac{\partial L}{\partial \mathbf{h}_i} = \mathbf{0}$, we obtain:

$$\sum_i \mathbf{x}_i \hat{\mathbf{h}}_i^{\mathsf{T}} = \hat{\mathbf{\Phi}} \mathbf{\Lambda}_M \quad \text{and} \quad \hat{\mathbf{\Phi}}^{\mathsf{T}} \mathbf{x}_i = \hat{\mathbf{h}}_i \implies \hat{\mathbf{h}}_i^{\mathsf{T}} = \mathbf{x}_i^{\mathsf{T}} \hat{\mathbf{\Phi}}$$

where we have used:

$$\frac{\partial}{\partial \mathbf{X}} \text{tr}\{\mathbf{X}\mathbf{B}\mathbf{X}^{\mathsf{T}}\} = \mathbf{X}\mathbf{B}^{\mathsf{T}} + \mathbf{X}\mathbf{B} \quad \text{and} \quad \frac{\partial \mathbf{a}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{b}}{\partial \mathbf{X}} = \mathbf{b}\mathbf{a}^{\mathsf{T}}.$$

- Therefore,

$$\sum_i \mathbf{x}_i \mathbf{x}_i^{\mathsf{T}} \hat{\mathbf{\Phi}} = \hat{\mathbf{\Phi}} \mathbf{\Lambda}_M \implies \mathbf{S}\hat{\mathbf{\Phi}} = \hat{\mathbf{\Phi}} \mathbf{\Lambda}_M \qquad (6)$$

- So, $\hat{\mathbf{\Phi}}$ comprises the M eigenvectors of $\mathbf{S}$.

Interpretation of $\mathbf{\Lambda}_M$

- Denote $\mathbf{X}$ as a $D \times N$ centered data matrix whose n-th column is given by $(\mathbf{x}_n - \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i)$.
- The projected data matrix is given by

$$\mathbf{Y} = \hat{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{X}$$

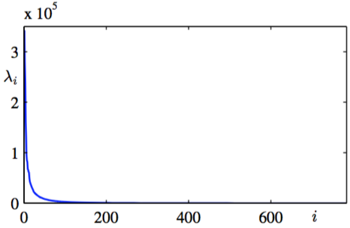
- The covariance matrix of the projected data is

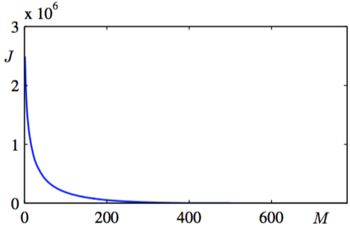
$$\begin{aligned}
\mathbf{Y}\mathbf{Y}^{\mathsf{T}} &= \left(\hat{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{X}\right)\left(\hat{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{X}\right)^{\mathsf{T}} \\
&= \hat{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{X}\mathbf{X}^{\mathsf{T}}\hat{\mathbf{\Phi}} \\
&= \hat{\mathbf{\Phi}}^{\mathsf{T}}\hat{\mathbf{\Phi}}\mathbf{\Lambda}_M \quad \text{(see the eigen-equation in Eq. 6)} \\
&= \mathbf{\Lambda}_M
\end{aligned}$$

- Therefore, the eigenvalues represent the variances of individual elements of the projected vectors.

Interpretation of Λ_M

- The eigenvalues are typically arranged in descending order: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$.
- This means that the first few principal components capture most of the variances.
- If we project $\mathbf{x}$ to M-dimensional space (i.e., keeping the first M PCs), the loss in variances is $J = \sum_{i=M+1}^{D} \lambda_i$.
- The variance "explained" by the first M PCs is $\sum_{i=1}^{M} \lambda_i$.





PCA on High-Dimensional Data

- When, the dimension D of $\mathbf{x}_i$ is very high, computing $\mathbf{S}$ and its eigenvectors directly are impractical.
- However, the rank of $\mathbf{S}$ is limited by the number of training examples: If there are N training examples, there will be at most $N-1$ eigenvectors with non-zero eigenvalues. If $N \ll D$, the principal components can be computed more easily.
- Let $\mathbf{X}$ be a data matrix comprising the mean-subtracted $\mathbf{x}_i$'s in its columns. Then, $\mathbf{S} = \mathbf{X}\mathbf{X}^\mathsf{T}$ and the eigen-decomposition of $\mathbf{S}$ is given by

$$\mathbf{S}\boldsymbol{\phi}_i = \mathbf{X}\mathbf{X}^\mathsf{T}\boldsymbol{\phi}_i = \lambda_i\boldsymbol{\phi}_i$$

- Instead of performing eigen-decomposition of $\mathbf{X}\mathbf{X}^\mathsf{T}$, we perform eigen-decomposition of

$$\mathbf{X}^\mathsf{T}\mathbf{X}\boldsymbol{\psi}_i = \lambda_i\boldsymbol{\psi}_i \tag{7}$$

- Pre-multipling both side of Eq. 7 by $\mathbf{X}$, we obtain

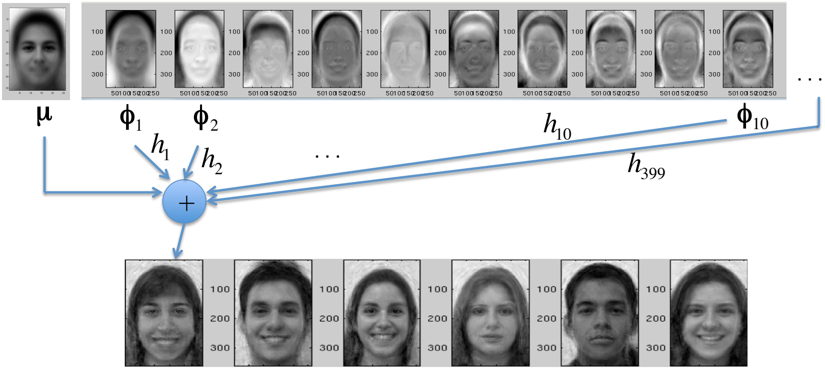
$$\mathbf{X}\mathbf{X}^{\mathsf{T}}(\mathbf{X}\boldsymbol{\psi}_i) = \lambda_i(\mathbf{X}\boldsymbol{\psi}_i)$$

- This means that if $\boldsymbol{\psi}_i$ is an eigenvector of $\mathbf{X}^{\mathsf{T}}\mathbf{X}$, then $\boldsymbol{\phi}_i = \mathbf{X}\boldsymbol{\psi}_i$ is an eigenvector of $\mathbf{S} = \mathbf{X}\mathbf{X}^{\mathsf{T}}$.
- So, all we need is to compute the $N-1$ eigenvectors of $\mathbf{X}^{\mathsf{T}}\mathbf{X}$, which has size $N \times N$.
- Note that $\boldsymbol{\phi}_i$ computed in this way is un-normalized. So, we need to normalize them by

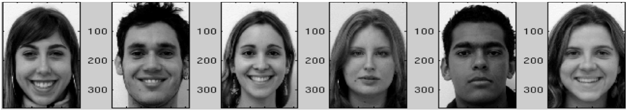
$$\boldsymbol{\phi}_i = \frac{\mathbf{X}\boldsymbol{\psi}_i}{\|\mathbf{X}\boldsymbol{\psi}_i\|}, \qquad i = 1, \ldots, N-1$$

Example Application of PCA: Eigenface

- Eigenface is one of the most well-known applications of PCA.



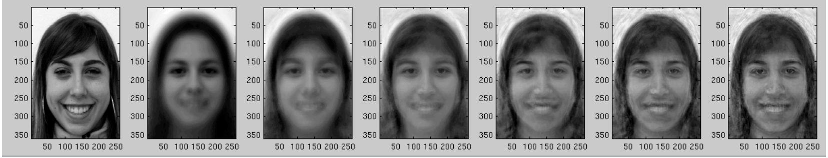
Reconstructed faces using 399 eigenfaces



Original faces

Example Application of PCA: Eigenface

- Faces reconstructed using different numbers of principal components (eigenfaces):

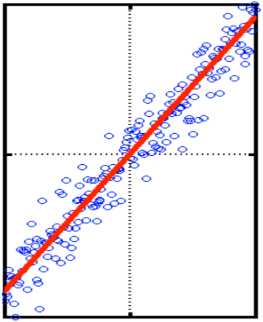


Original 1 PC 20 PCs 50 PCs 100 PCs 200 PCs 399 PCs

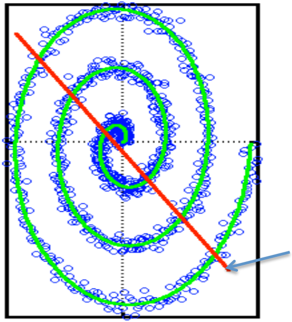
- See Lab2 of EIE4105 in http://www.eie.polyu.edu.hk/∼mwmak/myteaching.htm for implementation.

Limitations of PCA

- PCA will fail if the subspace is non-linear



Linear subspace (PCA is fine)

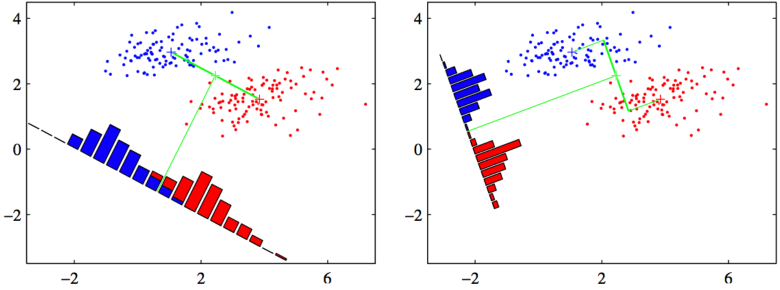


Nonlinear subspace (PCA fails)

- *Solution*: Use non-linear embedding such as ISOMAP or DNN

Fisher Discriminant Analysis

- FDA is a classification method to separate data into two classes.
- FDA could also be considered as a supervised dimension reduction method that reduces the dimension to 1.



Project data onto line joining the 2 means　　Project data onto FDA subspace

Fisher Discriminant Analysis

- The idea of FDA is to find a 1-D line so that the projected data give a **large separation** between the means of two classes while also giving a **small variance** within each class, thereby minimizing the class overlap.
- Assume that training data are projected onto a 1-D space using

$$y_n = \mathbf{w}^{\mathsf{T}}\mathbf{x}_n, \qquad n = 1, \ldots, N.$$

- Fisher criterion:

$$J(\mathbf{w}) = \frac{\text{Between-class scatter}}{\text{Within-class scatter}} = \frac{\mathbf{w}^{\mathsf{T}}\mathbf{S}_B\mathbf{w}}{\mathbf{w}^{\mathsf{T}}\mathbf{S}_W\mathbf{w}}$$

where

$$\mathbf{S}_B = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^{\mathsf{T}} \text{ and } \mathbf{S}_W = \sum_{k=1}^{2} \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathsf{T}}$$

are the *between-class* and *within-class* scatter matrices, respectively, and $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the class means.

Fisher Discriminant Analysis

- Note that only the direction of $\mathbf{w}$ matters. Therefore, we can always find a $\mathbf{w}$ that leads to $\mathbf{w}^{\mathsf{T}}\mathbf{S}_W\mathbf{w} = 1$.
- The maximization of $J(\mathbf{w})$ can be rewritten as:

$$\begin{array}{ll} \max_{\mathbf{w}} & \mathbf{w}^{\mathsf{T}}\mathbf{S}_B\mathbf{w} \\ \text{subject to} & \mathbf{w}^{\mathsf{T}}\mathbf{S}_W\mathbf{w} = 1 \end{array}$$

- The Lagrangian function is

$$L(\mathbf{w}, \lambda) = \frac{1}{2}\mathbf{w}^{\mathsf{T}}\mathbf{S}_B\mathbf{w} - \lambda(\mathbf{w}^{\mathsf{T}}\mathbf{S}_W\mathbf{w} - 1)$$

- Setting $\frac{\partial L}{\partial \mathbf{w}} = 0$, we obtain

$$\begin{aligned} & \mathbf{S}_B\mathbf{w} - \lambda\mathbf{S}_W\mathbf{w} = 0 \\ \Longrightarrow\ & \mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w} \\ \Longrightarrow\ & (\mathbf{S}_W^{-1}\mathbf{S}_B)\mathbf{w} = \lambda\mathbf{w} \end{aligned} \tag{8}$$

- So, $\mathbf{w}$ is the first eigenvector of $\mathbf{S}_W^{-1}\mathbf{S}_B$.

LDA on Multi-class Problems

- For multiple classes ($K > 2$ and $D > K$), we can use LDA to project D-dimensional vectors to M-dimensional vectors, where $1 < M < K$.
- $\mathbf{w}$ is extended to a matrix $\mathbf{W} = [\mathbf{w}_1 \ \cdots \ \mathbf{w}_M]$ and the projected scalar y_i is extended to a vector $\mathbf{y}_i$:

$$\mathbf{y}_n = \mathbf{W}^\mathsf{T}(\mathbf{x}_n - \boldsymbol{\mu}), \quad \text{where} \quad y_{nj} = \mathbf{w}_j^\mathsf{T}(\mathbf{x}_n - \boldsymbol{\mu}), \ j = 1, \ldots, M$$

  where $\boldsymbol{\mu}$ is the global mean of training vectors.
- The between-class and within-class scatter matrices become

$$\mathbf{S}_B = \sum_{k=1}^{K} N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^\mathsf{T}$$

$$\mathbf{S}_W = \sum_{k=1}^{K} \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^\mathsf{T}$$

  where N_k is the number of samples in the class k, i.e., $N_k = |\mathcal{C}_k|$.

LDA on Multi-class Problems

- The LDA criterion function:

$$J(\mathbf{W}) = \frac{\text{Between-class scatter}}{\text{Within-class scatter}} = \mathsf{Tr}\left\{ \left(\mathbf{W}^{\mathsf{T}}\mathbf{S}_B\mathbf{W}\right) \left(\mathbf{W}^{\mathsf{T}}\mathbf{S}_W\mathbf{W}\right)^{-1} \right\}$$

- Constrained optimization:

$$\begin{array}{rl} \max_{\mathbf{W}} & \mathsf{Tr}\{\mathbf{W}^{\mathsf{T}}\mathbf{S}_B\mathbf{W}\} \\ \text{subject to} & \mathbf{W}^{\mathsf{T}}\mathbf{S}_W\mathbf{W} = \mathbf{I} \end{array}$$

  where $\mathbf{I}$ is an $M \times M$ identity matrix.

- Note that unlike PCA in Eq. 5, because of the matrix $\mathbf{S}_W$ in the constraint, we need to find one $\mathbf{w}_j$ at a time.
- Note also that the constraint $\mathbf{W}^{\mathsf{T}}\mathbf{S}_W\mathbf{W} = \mathbf{I}$ suggests that $\mathbf{w}_j$'s may not be orthogonal to each other [2].

LDA on Multi-class Problems

- To find $\mathbf{w}_j$, we write the Lagrangian function as:

$$L(\mathbf{w}_j, \lambda_j) = \mathbf{w}_j^{\mathsf{T}} \mathbf{S}_B \mathbf{w}_j - \lambda_j (\mathbf{w}_j^{\mathsf{T}} \mathbf{S}_W \mathbf{w}_j - 1)$$

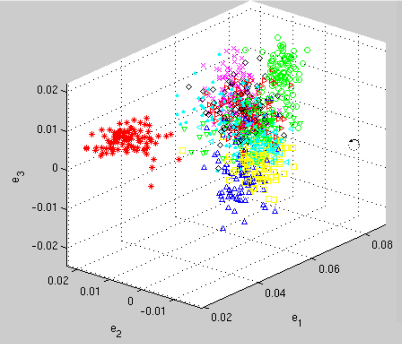
- Using Eq. 8, the optimal solution of $\mathbf{w}_j$ satisfies

$$(\mathbf{S}_W^{-1} \mathbf{S}_B) \mathbf{w}_j = \lambda_j \mathbf{w}_j$$

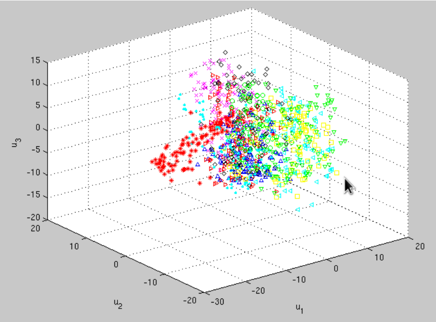
- Therefore, $\mathbf{W}$ comprises the first M eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$. A more formal proof can be find in [1].
- As the maximum rank of $\mathbf{S}_B$ is $K-1$, $\mathbf{S}_W^{-1}\mathbf{S}_B$ has at most $K-1$ non-zero eigenvalues. As a result, M can be at most $K-1$.
- After the projection, the vectors $\mathbf{y}_n$'s can be used to train a classifier (e.g., SVM) for classification.

PCA vs. LDA

- Project 784-dim vectors derived from 28×28 handwritten digits to 3-D space:

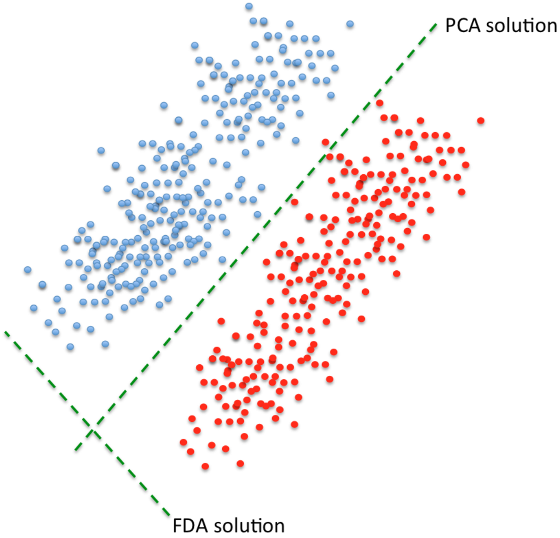


LDA



PCA

PCA vs. LDA



References

[1] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. San Diego, California, USA: Academic Press.

[2] Luo, Dijun Luo, Ding, Chris, and Huang, Heng (2011) "Linear Discriminant Analysis: New Formulations and Overfit Analysis", *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*.