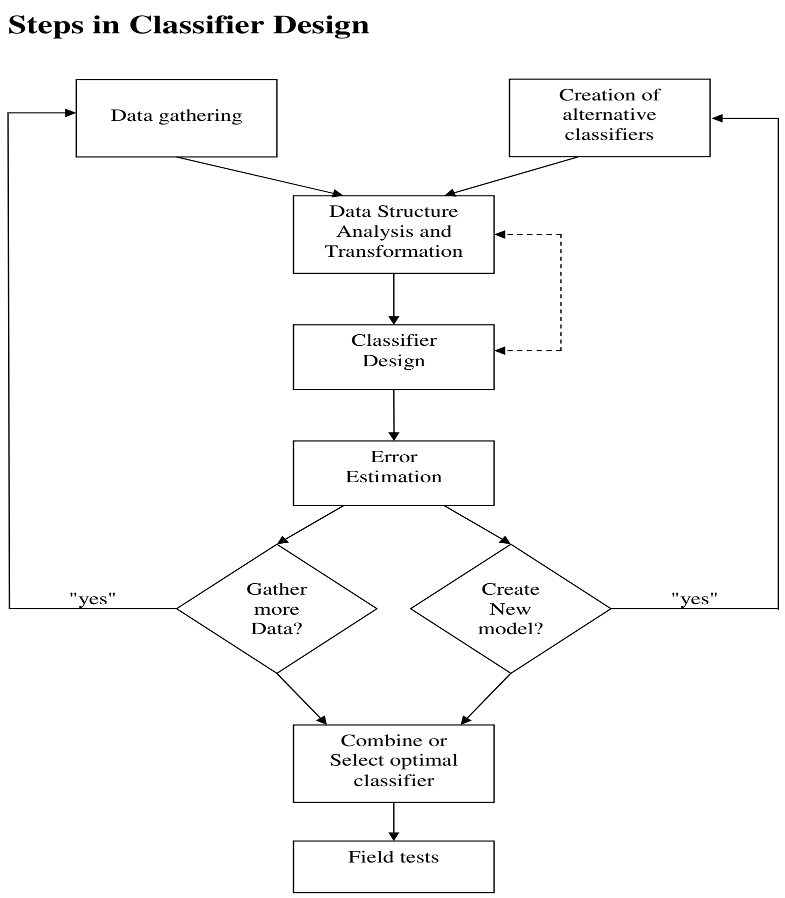
Reminder: Basic Setup



Feature Extraction

Basic Issue: Given a feature set $\boldsymbol{x} = (x_1, \ldots, x_d)$, find a transformation leading to the best class separability.

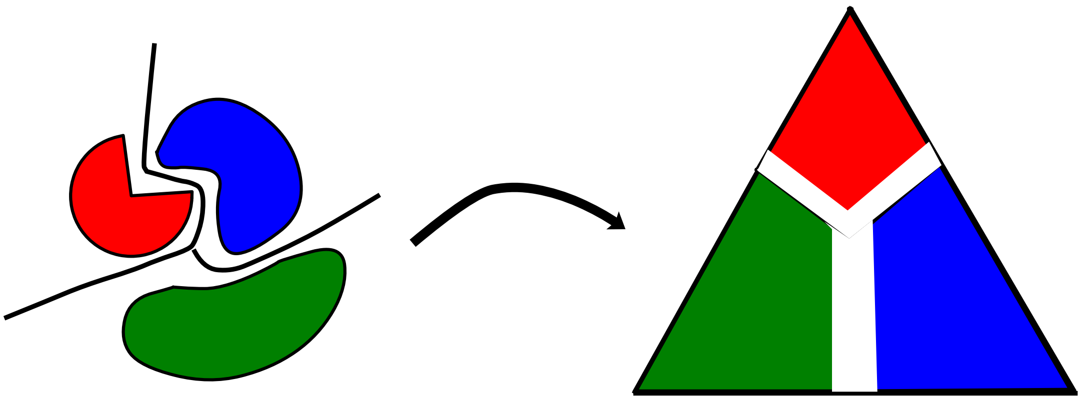
Optimal solution: The Bayes discriminants functions!

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \mapsto g_1(\boldsymbol{x}), \ldots, g_C(\boldsymbol{x})$$

$$g_c(\boldsymbol{x}) = \log P(\omega_c|\boldsymbol{x})$$

But **infeasible** in reality

Tradeoff The border-line between feature extraction and classification is blurred



FEATURE EXTRACTION - DEFINITION

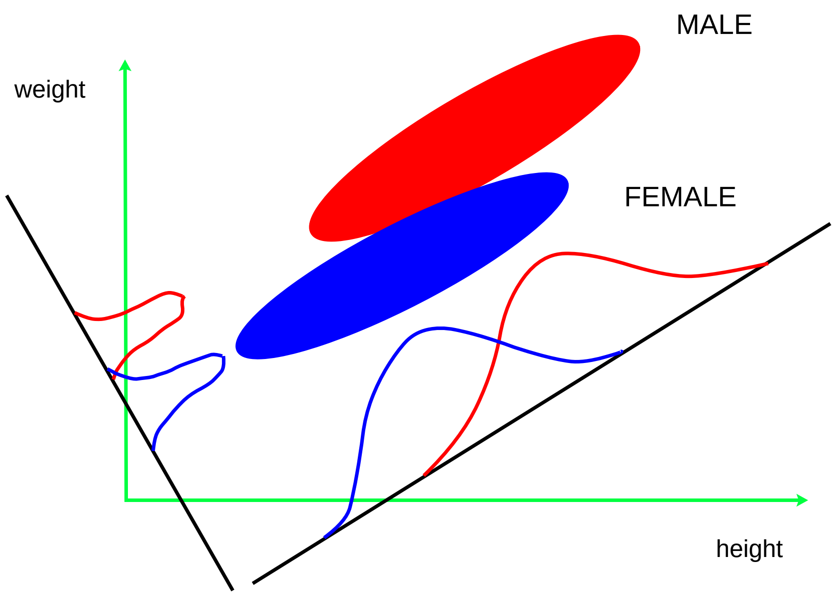
Feature extraction

$$(x_1, x_2, \ldots, x_d) \mapsto (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))$$

Maintain good class-separability

Feature selection Select a **subset** of 'good' features: $x_1, \ldots, x_d \mapsto x_{i_1}, \ldots, x_{i_k}$.

Selection vs. extraction: Individual features may be bad, but combination good



Feature Extraction - Motivation

Note: Information is lost in transformation, but can gain in:

Computation Reduce number of parameters, leading to simpler algorithmic implementation

Statistical error For finite data sets can get performance enhancements

- ⋆ Reduction of estimation error ('bias/variance tradeoff')

Visualization

Strategies:

Unsupervised Disregard class information

Supervised Use class information

PCA - Motivation

PCA - Principal Component Analysis

Unsupervised: Disregards label information

Dimensionality reduction: Project feature vector $\boldsymbol{x}$ on to low-dimensional (linear) space, retaining 'as much information as possible'

PCA **Objective:** Find MSE-optimal m-dim. linear representation of d-dim signal, $m \leq d$.

PCA - Derivation I

Input: $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{x} \sim p(\boldsymbol{x})$ (or data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$)

Output: A 'good' m-dimensional representation $\hat{\boldsymbol{x}}$

$$\boldsymbol{x}_{(d\times 1)} \mapsto \tilde{\boldsymbol{x}}_{(m\times 1)} \mapsto \hat{\boldsymbol{x}}_{(d\times 1)}$$

Criterion: Minimize $\mathbf{E}_m = \mathbf{E}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$

Formalization:

$$\boldsymbol{x} = \sum_{i=1}^{d} x^{(i)} \mathbf{u}_i \qquad ; \qquad x^{(i)} = \boldsymbol{x}^\top \mathbf{u}_i$$

$$\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij} \qquad \text{(Orthonormal basis)}$$

m-dimensional representation:

$$\tilde{\boldsymbol{x}} = \sum_{i=1}^{m} x^{(i)} \mathbf{u}_i + \sum_{i=m+1}^{d} b^{(i)} \mathbf{u}_i$$

$$= m - \text{projection} + \text{residual}$$

Question: Which m-dimensional subspace?

PCA - Derivation II

Squared loss: Set $x^{(i)} = \boldsymbol{x}^\top \mathbf{u}_i$,

$$E_m = \mathbf{E}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2 = \mathbf{E}\left\{\sum_{i=m+1}^{d} (x^{(i)} - b^{(i)})^2\right\}$$

Minimum at: (Set gradient to 0)

$$b^{(i)} = \mathbf{u}_i^\top \mathbf{E}[\boldsymbol{x}]$$

Conclude

$$E_m = \sum_{i=m+1}^{d} \mathbf{E}\left\{\mathbf{u}_i^\top(\boldsymbol{x} - \mathbf{E}\boldsymbol{x})\right\}^2$$

$$= \sum_{i=m+1}^{d} \mathbf{u}_i^\top Q \mathbf{u}_i$$

$$Q = \mathbf{E}\left[(\boldsymbol{x} - \mathbf{E}\boldsymbol{x})(\boldsymbol{x} - \mathbf{E}\boldsymbol{x})^\top\right]$$

Remaining question: Selection of optimal $\mathbf{u}_i$

Q - symmetric, non-negative definite

Assume: From now $\mathbf{E}[\boldsymbol{x}] = \mathbf{0}$

PCA - Optimal Basis I

Optimization problem:

$$\min_{\mathbf{u}_i} \left\{ \frac{1}{2} \sum_{i=m+1}^{d} \mathbf{u}_i^\top Q \mathbf{u}_i \right\}$$
$$\text{s.t.} \quad \mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$$

Lagrangian:

$$\mathcal{L}(\{\mathbf{u}_i\}) = \frac{1}{2} \sum_{i=m+1}^{d} \mathbf{u}_i^\top Q \mathbf{u}_i - \frac{1}{2} \sum_{i,j=m+1}^{d} \sum^{d} \mu_{ij} (\mathbf{u}_i^\top \mathbf{u}_j - \delta_{ij})$$

Set $\frac{\partial \mathcal{L}(\{\mathbf{u}_i\})}{\partial \mathbf{u}_i} = 0$

$$Q\mathbf{u}_i = \sum_j \mu_{ij} \mathbf{u}_j \qquad (i = m+1, \ldots, d)$$

PCA - Optimal Basis II

Set

$$M \sim ((d-m)\times(d-m)), \quad (M)_{ij} = \mu_{ij}$$
$$U \sim (d \times (d-m)), \quad \text{Columns} = \mathbf{u}_i$$

Matrix notation:

$$QU = UM \quad \Rightarrow \quad U^\top QU = M$$

Possible solution:

$$\mathbf{u}_i = i\text{'th eigenvector of } Q$$
$$M = \text{diagonal eigenvalue matrix}$$

PCA - Non Uniqueness

Recall

$$U^\top Q U = M \qquad (*)$$

Generate new solution: Set Ψ orthogonal matrix

$$\tilde{M} = \Psi M \Psi^\top$$

$$\tilde{U} = U \Psi^\top \quad \text{(rotation)}$$

Thus $M = \Psi^\top \tilde{M} \Psi$, from (*)

$$\tilde{U}^\top Q \tilde{U} = \tilde{M}$$

Conclude: If U, M are solutions, so are $\tilde{U}, \tilde{M}$

Decorrelation: Set

$$\text{columns}(U) = \text{eigenvectors}$$

$$y_i = \mathbf{u}_i^\top \boldsymbol{x}$$

$$\mathbf{E}[y_i y_j] = \mathbf{u}_i^\top Q \mathbf{u}_j = \lambda_i \delta_{ij}$$

Note that $\mathbf{E}[x_i x_j]$ is not diagonal.

PCA Properties

Q is positive semi-definite Implying

$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0.$

Error: Lowest if $\mathbf{u}_i$ are the *leading eigenvectors*, since

$$E_m = \sum_{i=m+1}^{d} \mathbf{u}_i^\top Q \mathbf{u}_i$$

$$= \sum_{i=m+1}^{d} \lambda_i$$

PCA algorithm:

- Compute covariance matrix Q and its eigenvectors and eigenvalues
- Construct m-dimensional approximation

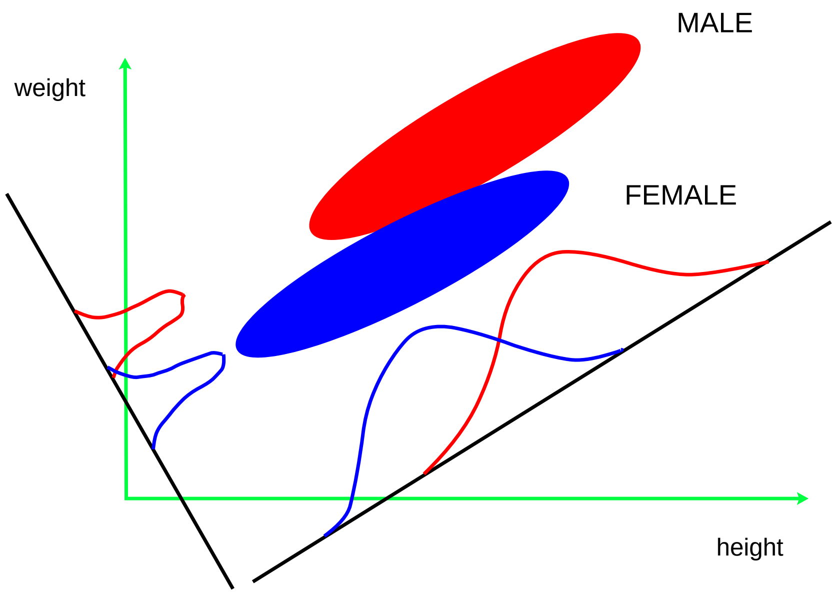
$$\hat{\boldsymbol{x}} = \sum_{i=1}^{m} x^{(i)} \mathbf{u}_i \qquad \left(x^{(i)} = \boldsymbol{x}^\top \mathbf{u}_i \right)$$

PCA for Classification

Reduced feature vector: Replace $\boldsymbol{x} \in \mathbb{R}^d$ by $(\mathbf{u}_1^\top \boldsymbol{x}, \ldots, \mathbf{u}_m^\top \boldsymbol{x})$ as input to classifier

Optimality: No classification optimality is achieved. The optimality is purely representational, using MSE.

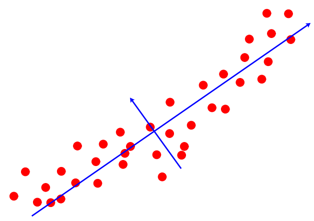
- ⋆ This can actually be harmful



PCA - High Variance Projection

Definition:

- First PC, $\mathbf{v}_1$ - direction of largest variance
- kth PC, $\mathbf{v}_k$ - direction of maximal variance, orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}$



Theorem The k-th PC is the normalized eigenvector $\mathbf{v}_k$ corresponding to the eigenvalue λ_k of Q, where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$.

Principal Components - Proof

k = 1 Consider $\mathbf{p} \in \mathbb{R}^d$ with $\|\mathbf{p}\| = 1$ ($\mathbf{E}[\boldsymbol{x}] = \mathbf{0}$)

$$\sigma_1^2 = \mathbf{E}(\mathbf{p}^\top \boldsymbol{x})^2 = \mathbf{E}(\mathbf{p}^\top \boldsymbol{x}\boldsymbol{x}^\top \mathbf{p}) = \mathbf{p}^\top Q \mathbf{p}$$

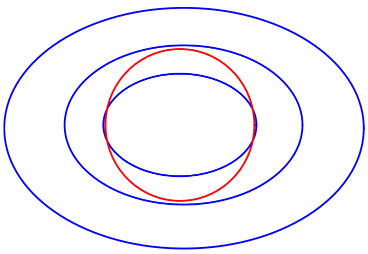
Set $Q = \sum_{i=1}^d \lambda_i \mathbf{v}_i \mathbf{v}_i^\top$ (spectral representation),

$$\sigma_1^2 = \sum_{i=1}^d \lambda_i (\mathbf{p}^\top \mathbf{v}_i)^2$$

Set $\mathbf{p} = \sum_j a_j \mathbf{v}_j \quad \Rightarrow \quad \sigma_1 = \sum_i \lambda_i a_i^2$

Solve: $$\max_{\mathbf{a}} \left\{ \sum_{i=1}^d \lambda_i a_i^2 \right\} \qquad \text{s.t.} \ \sum_{i=1}^d a_i^2 = 1$$

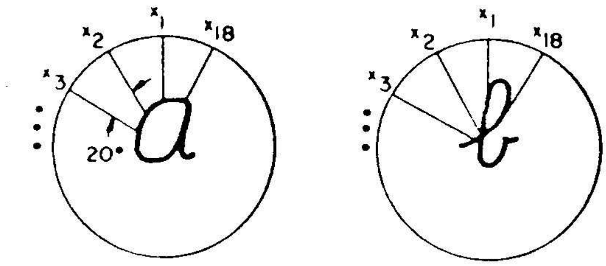
maximum for $a_1 = \pm 1$, $\mathbf{p} = \pm \mathbf{v}_1$ and $\sigma_1 = \lambda_1$.

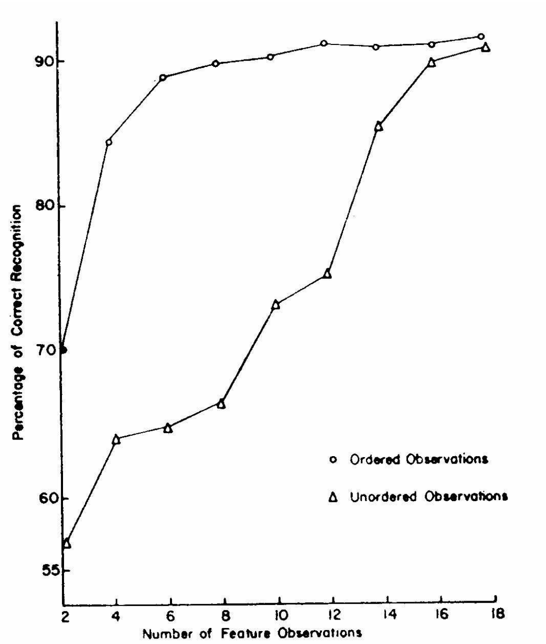


k > 1 By induction, using $\mathbf{p}_k^\top \mathbf{v}_i$, $i = 1, \ldots, k-1$, obtain result.

Application Example

- Hand-writing recognition (a,b,c,d)
- Input vector - 18 dimensions
- 240 samples





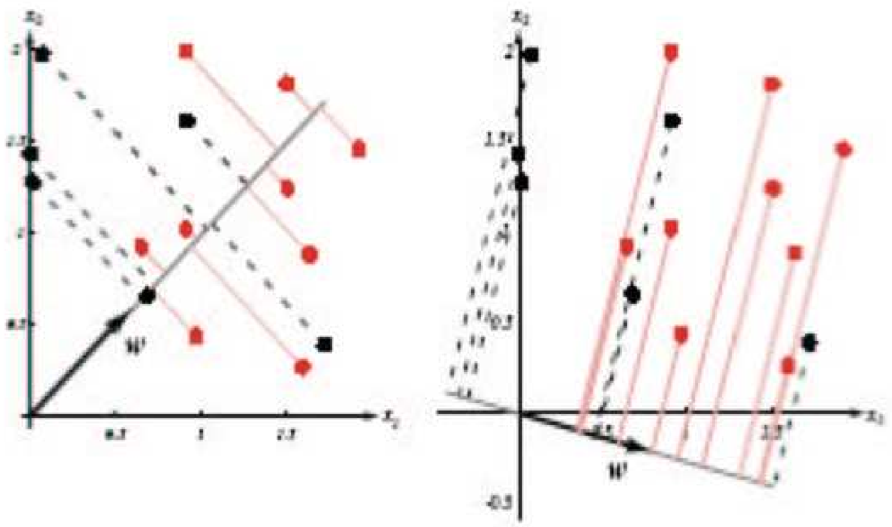
PCA - General Comments

- Provide optimal reconstruction - linear mapping, quadratic loss.
- Not robust (outliers) - variations exist
- Data reduction as pre-processor for supervised learning
- Nonlinear PCA possible - analysis hard
- On-line versions available
- A major problem for classification is that PCA disregards the labels (unsupervised)

The Fisher Discriminant I

Objective: Find one-dimensional projection $\boldsymbol{w}^\top \boldsymbol{x}$ which achieves *'best separability'* of the classes

Labels Here label information is clearly used



The Fisher Discriminant II

Mean and scatter matrices:

$$\mathbf{m}_c = \frac{1}{n_c} \sum_{\boldsymbol{x} \in \mathcal{X}_c} \boldsymbol{x}$$

$$S_c = \sum_{\boldsymbol{x} \in \mathcal{X}_c} (\boldsymbol{x} - \mathbf{m}_c)(\boldsymbol{x} - \mathbf{m}_c)^\top$$

Transformed mean and scatter:

$$y = \boldsymbol{w}^\top \boldsymbol{x}$$

$$\tilde{m}_c = \frac{1}{n_c} \sum_{y \in \mathcal{Y}_c} y = \boldsymbol{w}^\top \mathbf{m}_c$$

$$\tilde{s}_c^2 = \sum_{y \in \mathcal{Y}_c} (y - \tilde{m}_c)^2$$

Objective to maximize: (heuristic!)

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

The Fisher Discriminant III

Within and between class scatter:

$$S_W = S_1 + S_2$$

$$S_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top \qquad \text{(rank one)}$$

$$\begin{aligned}
\tilde{s}_c^2 &= \sum_{\boldsymbol{x} \in \mathcal{X}_c} (\boldsymbol{w}^\top \boldsymbol{x} - \boldsymbol{w}^\top \mathbf{m}_c)^2 \\
&= \sum_{\boldsymbol{x} \in \mathcal{X}_c} \boldsymbol{w}^\top (\boldsymbol{x} - \mathbf{m}_c)(\boldsymbol{x} - \mathbf{m}_c)^\top \boldsymbol{w} \\
&= \boldsymbol{w}^\top S_c \boldsymbol{w}
\end{aligned}$$

$$\begin{aligned}
(\tilde{m}_1 - \tilde{m}_2)^2 &= (\boldsymbol{w}^\top \mathbf{m}_1 - \boldsymbol{w}^\top \mathbf{m}_2)^2 \\
&= \boldsymbol{w}^\top S_B \boldsymbol{w}
\end{aligned}$$

Objective: Maximize

$$\boxed{J(\boldsymbol{w}) = \frac{\boldsymbol{w}^\top S_B \boldsymbol{w}}{\boldsymbol{w}^\top S_W \boldsymbol{w}}}$$

Note: $\|\boldsymbol{w}\|$ immaterial

The Fisher Discriminant IV

Require

$$\max_{\boldsymbol{w}} J(\boldsymbol{w}) = \max_{\boldsymbol{w}} \left\{ \frac{\boldsymbol{w}^\top S_B \boldsymbol{w}}{\boldsymbol{w}^\top S_W \boldsymbol{w}} \right\}$$

Set $\partial J(\boldsymbol{w})/\partial \boldsymbol{w} = \mathbf{0}$, obtaining

$$\begin{aligned} S_B \boldsymbol{w} &= \left(\frac{\boldsymbol{w}^\top S_B \boldsymbol{w}}{\boldsymbol{w}^\top S_W \boldsymbol{w}} \right) S_W \boldsymbol{w} \\ &= \lambda S_W \boldsymbol{w} \qquad \text{(generalized e.v. problem)} \end{aligned}$$

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

Since scale is irrelevant and

$$S_B \boldsymbol{w} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^\top \boldsymbol{w} \propto (\mathbf{m}_1 - \mathbf{m}_2)$$

$$\boxed{\boldsymbol{w}^* = S_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)}$$

Fisher: Label information used!

PCA: Covariance of unlabelled data used.

Supervised Feature Extraction

Motivation: Extend Fisher to multiple dimensions

Criterion: Need a 'simple' class separability criterion

Recall:

$$P_c = \frac{n_c}{n}$$

$$\mathbf{m}_c = \frac{1}{n_c} \sum_{X_k \in C_c} \boldsymbol{x}_k$$

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \boldsymbol{x}_k$$

$$\Sigma_c = \frac{1}{n_c} \sum_{X_k \in \mathcal{X}_c} (\boldsymbol{x}_k - \mathbf{m}_c)(\boldsymbol{x}_k - \mathbf{m}_c)^\top$$

$$\Sigma = \frac{1}{n} \sum_{k=1}^{n} (\boldsymbol{x}_k - \mathbf{m})(\boldsymbol{x}_k - \mathbf{m})^\top$$

Separability Criteria

Source: Fukunaga, Chapter 10

Scatter matrices

$$S_w = \sum_{c=1}^{C} P_c \Sigma_c$$

$$S_b = \sum_{c=1}^{C} P_c (\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^\top \quad \text{(rank } C-1\text{)}$$

$$S_m = S_w + S_b$$

Basic idea: Find projection directions which maximize 'separation' between classes

Separability criteria: (for example)

$$J_1 = \mathrm{Tr}(S_2^{-1} S_1) \qquad (\text{e.g. }, S_1 = S_b, S_2 = S_w)$$

$$J_2 = \log \left| S_2^{-1} S_1 \right|$$

Note: S_b cannot be used in J_2 - rank $C - 1$

Solution

Notation: S_c - one of S_w, S_b and S_w

$$\boldsymbol{y} = A^\top \boldsymbol{x} \quad (\boldsymbol{x} \in \mathbb{R}^d,\ \boldsymbol{y} \in \mathbb{R}^m)$$

$$S_{iy} = A^\top S_{ix} A$$

Solution: (skip math in class - see next slides)

$$J_1(A^*) = \mu_1 + \mu_2 + \cdots + \mu_m$$

$$\{\mu_i\}_{i=1}^m = \text{largest eigenvalues of of } S_{2x}^{-1} S_{1x}$$

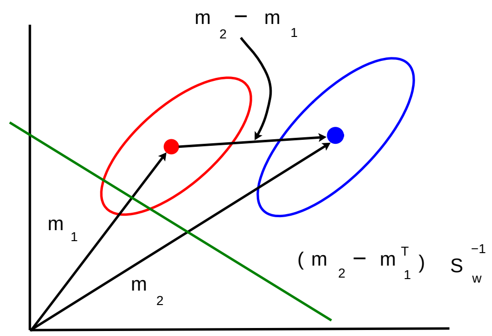
Conclusion: Maximal separation achieved projecting onto m eigenvectors corresponding to **largest eigenvalues** of $S_{2x}^{-1} S_{1x}$

Compare to PCA

Same form of solution, except that $S_2^{-1}S_1$ used rather than covariance matrix $\mathbf{E}[\boldsymbol{x}\boldsymbol{x}^\top]$.

Two-class problem: $J_1 = \mathrm{Tr}\left(S_w^{-1}S_b\right)$

$$y = (\mathbf{m}_2 - \mathbf{m}_1)^\top S_w^{-1}\boldsymbol{x}$$



M Whitening Transformation

Motivation: Will use below

Drawbacks: Outliers, may destroy structure

Assume:

$$\mathbf{E}[\boldsymbol{x}] = \mathbf{0}$$

$$Q = \mathbf{E}[\boldsymbol{x}\boldsymbol{x}^\top]$$

$$\Phi^\top Q \Phi = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$$

$$\Phi = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_d) \qquad \text{(eigenvectors)}$$

Orthogonal transformation: $\boldsymbol{y} = \Phi^\top \boldsymbol{x}$

Whitening transformation: **Not orthogonal!**

$$\boldsymbol{y} = \Lambda^{-1/2}\Phi^\top \boldsymbol{x}$$

$$\begin{aligned} Q_y &= \Lambda^{-1/2}\Phi^\top Q \Phi \Lambda^{-1/2} \\ &= I \end{aligned}$$

M SIMULTANEOUS DIAGONALIZATION I

Objective: Simultaneously diagonalize 2 symmetric matrices Σ_1 and Σ_2

Θ, Φ eigenvalue/eigenvector matrices of Σ_1

1. Whiten Σ_1

$$\boldsymbol{y} = \Theta^{-1/2}\Phi^\top \boldsymbol{x}$$

Then

$$\Theta^{-1/2}\Phi^\top \Sigma_1 \Phi\Theta^{-1/2} = I$$

$$\Theta^{-1/2}\Phi^\top \Sigma_2 \Phi\Theta^{-1/2} = K \quad \text{(not diagonal)}$$

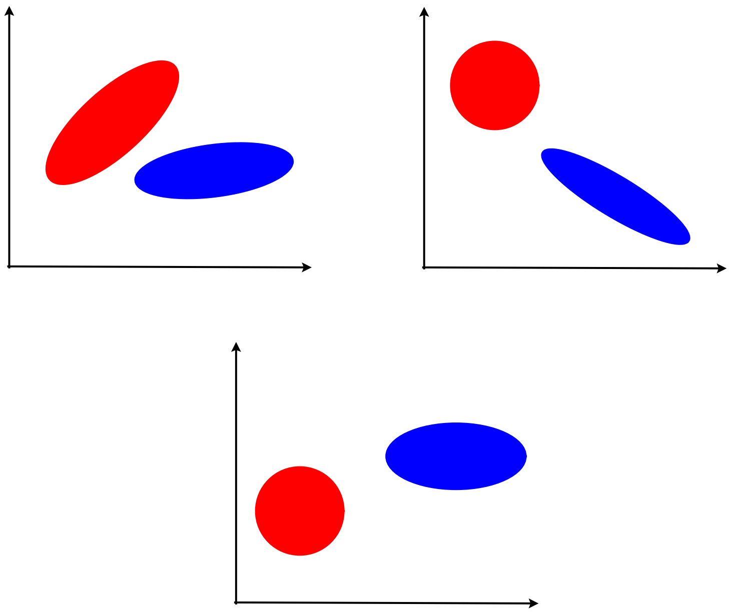
2. Diagonalize Σ_2 (unit matrix is invariant)
Note: K is symmetric

$$\mathbf{z} = \Psi^\top \boldsymbol{y}$$

$$\Psi^\top I \Psi = I$$

$$\Psi^\top K \Psi = \Lambda$$

M Simultaneous Diagonalization II



Theorem: (see Fukunaga pp. 31-33, distributed)

$$A^\top \Sigma_1 A = I \qquad \& \qquad A^\top \Sigma_2 A = \Lambda$$

where

$$\left(\Sigma_1^{-1}\Sigma_2\right) A = A\Lambda \qquad (*)$$

Note: $\lambda_1, \lambda_2, \ldots, \lambda_d$ are eigenvalues of $\Sigma_1^{-1}\Sigma_2$

M Linear Transformation I

Notation: S_c - one of S_w, S_b and S_w

$$\boldsymbol{y} = A^\top \boldsymbol{x} \quad (\boldsymbol{x} \in \mathbb{R}^d,\ \boldsymbol{y} \in \mathbb{R}^m)$$

$$S_{iy} = A^\top S_{ix} A$$

Objective: Find A which optimizes J in the y-space

Optimization of J_1

$$\begin{aligned} J_1(A) &= \mathrm{Tr}(S_{2y}^{-1} S_{1y}) \\ &= \mathrm{Tr}\left[\left(A^\top S_{2x} A\right)^{-1} \left(A^\top S_{1x} A\right)\right] \end{aligned}$$

Taking derivative w.r.t. A, setting to 0

$$\frac{\partial J_1(A)}{\partial A} = -2 S_{2x} A S_{2y}^{-1} S_{1y} S_{2y}^{-1} + 2 S_{1x} A S_{2y}^{-1} = 0$$

(Use matrix derivatives manual - see course webpage under auxiliary resources)

Recall $S_{2y} = A^\top S_{2x} A$

M Separability Criteria II

$$\left(S_{2x}^{-1}S_{1x}\right)A = A\left(S_{2y}^{-1}S_{1y}\right) \qquad (*)$$

Simultaneously diagonalize S_{1y} and S_{2y} to $\boldsymbol{\mu}$ and I (by whitening),

$$\mathbf{z} = B^\top \boldsymbol{y} \quad (\mathbf{z} \in \mathbb{R}^m)$$

$$B^\top S_{1y} B = \boldsymbol{\mu} \quad ; \quad B^\top S_{2y} B = I \qquad (**)$$

From $(**)$ easily find that $S_{2y}^{-1}S_{1y} = B\boldsymbol{\mu}B^{-1}$

Substituting on r.h.s. of $(*)$

$$\boxed{\left(S_{2x}^{-1}S_{1x}\right)(AB) = (AB)\boldsymbol{\mu} \qquad (\#)}$$

Observe:

- ⋆ $\mu_1, \ldots, \mu_m$ eigenvalues of $S_{2y}^{-1}S_{1y}$ (see 7.23)
- ⋆ From $(\#)$, $\mu_1, \ldots, \mu_m$ eigenvalues of $S_{2x}^{-1}S_{1x}$ as well

M Invariance of Criterion

Claim: $\mathrm{Tr}\left(S_{2z}^{-1}S_{1z}\right) = \mathrm{Tr}\left(S_{2y}^{-1}S_{1y}\right)$

Proof:

$$\begin{aligned}
\mathrm{Tr}\left(S_{2z}^{-1}S_{1z}\right) &= \mathrm{Tr}\left\{\left(B^\top S_{2y}B\right)^{-1}\left(B^\top S_{1y}B\right)\right\} \\
&= \mathrm{Tr}\left(B^{-1}S_{2y}^{-1}\left(B^\top\right)^{-1}B^\top S_{1y}B\right) \\
&= \mathrm{Tr}\left(S_{2y}^{-1}S_{1y}BB^{-1}\right) \\
&= \mathrm{Tr}\left(S_{2y}^{-1}S_{1y}\right)
\end{aligned}$$

Used $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$.

M Separability Criteria III

Recall

$$
\begin{aligned}
J_1(A) &= \mathrm{Tr}\left(S_{2y}^{-1} S_{1y}\right) \\
&= \mu_1 + \mu_2 + \cdots + \mu_m
\end{aligned}
$$

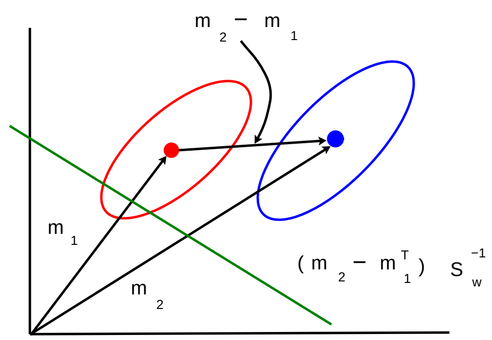
Conclusion: Maximal separation achieved projecting onto m eigenvectors corresponding to **largest eigenvalues** of $S_{2x}^{-1} S_{1x}$

Compare to PCA:

Same form of solution, except that $S_2^{-1} S_1$ used rather than covariance matrix $\mathbf{E}[\boldsymbol{x}\boldsymbol{x}^\top]$.

Two-class problem: $J_1 = \mathrm{Tr}\left(S_w^{-1} S_b\right)$

$$
y = (\mathbf{m}_2 - \mathbf{m}_1)^\top S_w^{-1} \boldsymbol{x}
$$



M Linear Transformation II

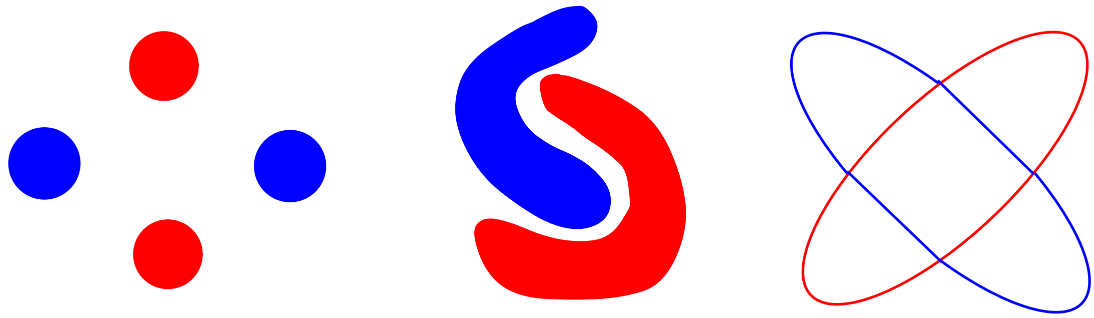
Observe:

- ⋆ S_{1x} and S_{2x} are symmetric, but $S_{2x}^{-1}S_{1x}$ is not necessarily symmetric
- ⋆ Eigenvalues and eigenvectors of $S_{2x}^{-1}S_{1x}$ obtained by simultaneous diagonalization of S_{1x} and S_{2x}.
  - Eigenvalues real and positive
  - Eigenvectors real and orthogonal w.r.t. S_{2x}

Separability Criteria IV

Caveat: Above linear procedures only effective for unimodal and weakly-overlapping class conditional distributions

What about the following situations?



Need more refined, nonlinear approaches, e.g.

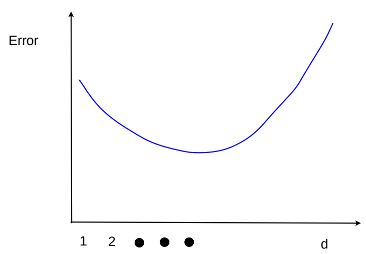
- ⋆ Self organizing maps
- ⋆ Nonlinear PCA (several variants)
- ⋆ Manifold embedding and eigenmaps
- ⋆ Kernel methods (discuss later)

Comment Before discussing some of these, briefly discuss feature subset selection

Feature Subset Selection I

Objective: Select a subset of features leading to best classification

Expectation: Optimal subset exists, due to bias/variance balance for finite sample



Caveat:

- ⋆ Combinatorial explosion - 2^d subsets
- ⋆ Cannot directly evaluate true error - estimates are noisy

Solution:

- ⋆ Greedy algorithms
- ⋆ Simplified criteria

Feature Subset Selection II

Simplified criteria Usually use separability criteria (based on covariance matrices)

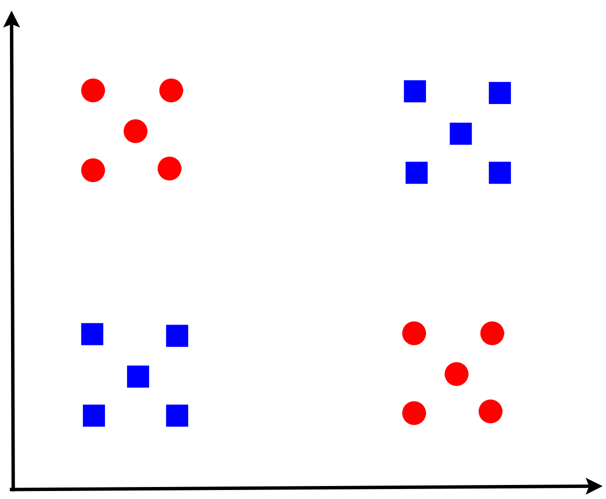
- ⋆ Observe: Monotinicity

$$J(X^+) \geq J(X) \qquad X^+ = X \cup x$$

Problem: Cannot select optimal number of features

Still useful: Compare subsets of the same size

Note: Discarding features can be dangerous!



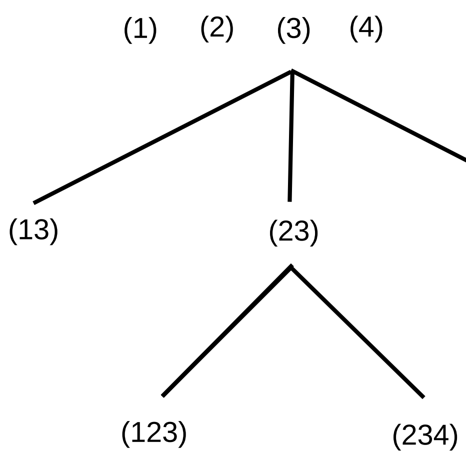
Sequential procedures I

Simple idea: Select k features which are individually best

- ⋆ Discards correlations
- ⋆ Features may be individually poor, but excellent in combination (Recall 7.33)

Forward selection: (greedy)

- ⋆ Consider all features individually and select one leading to maximal criterion
- ⋆ Add successive feature that yields largest increase in criterion

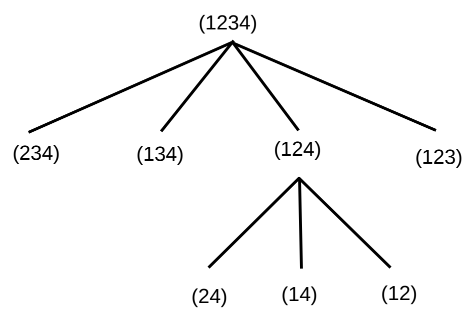


Sequential procedures II

Difficulty with forward selection: Situation as in 7.33, where two features are good, but each is poor

Backward elimination:

- ⋆ Initialize to full set of features
- ⋆ Sequentially eliminate a feature leading to minimal reduction in the value of the criterion



Computation: Heavier than forward selection (consider larger subsets)

Extensions: e.g., At kth stage add ℓ features and eliminate r

Projection Pursuit I

Reference: Ripley, Chap. 9.1, and Friedman JASA, 82: 249-266, 1987

Motivation: PCA looks for structure in the variance - insensitive to 'clumping' structure

Basic idea: Find directions maximizing a 'measure of interestingness'

- ⋆ Diaconis and Freedman ('84) - a random projection of high-dim data is similar to a sample from a multivariate Gaussian

Interstingness: Measure deviation from normality

Projection Pursuit II

To achieve affine invariance whiten the data:

- ⋆ Transform to (robust) principal components
- ⋆ Discard directions with small variance
- ⋆ Rescale each component to unit variance

Deviation from normality indices: Let $\hat{f}_n$ be a density estimate (e.g., using kernels):

$$I_2(\boldsymbol{w}) = \sum_{i=1}^{n} \left(\hat{f}_n(\boldsymbol{w}^\top \boldsymbol{x}_c) - \phi(\boldsymbol{w}^\top \boldsymbol{x}_c) \right)^2$$

$$I_{\mathrm{KL}}(\boldsymbol{w}) = \sum_{i=1}^{n} D_{\mathrm{KL}}[\hat{f}_n(\boldsymbol{w}^\top \boldsymbol{x}_c) \| \phi(\boldsymbol{w}^\top \boldsymbol{x}_c)]$$

Extensions: Multivariate projections

Difficulties:

- Criterion selection
- Robustness
- Computational burden

Nonlinear PCA

Website:

http://www.iro.umontreal.ca/ kegl/research/pcurves/

Motivation Standard PCA cannot capture non-linear structure

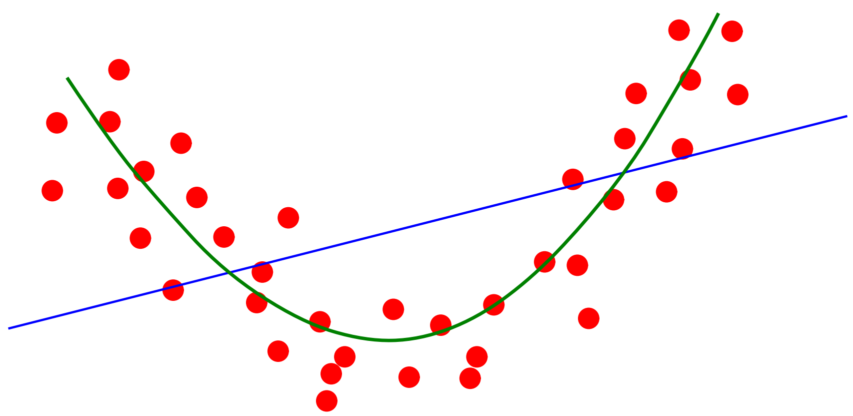


Figure 1:

Nonlinear PCA Several versions exist:

- Neural network
- Local PCA
- Self-organizing map
- Kernel PCA

Nonlinear PCA

Basic Idea:

PCA: Can phrase as follows:

Find line for which

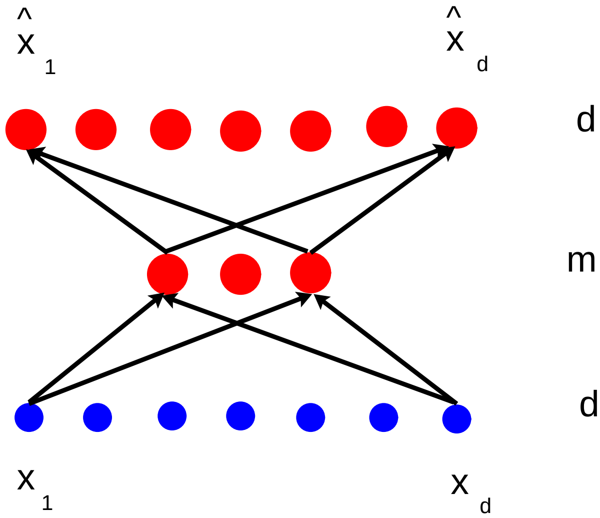
$$\sum_{i=1}^{n} \mathrm{dist}(\boldsymbol{x}_i, \mathrm{line})^2 \quad \text{is minimal}$$

Nonlinear PCA: Find curve for which distance is minimized

$$\sum_{i=1}^{n} \mathrm{dist}(\boldsymbol{x}_i, \mathrm{curve})^2 \quad \text{is minimal}$$

But, must restrict the irregularity of the curve, e.g., length of the line

Neural Network PCA I



Autossociative 2 layer network

Output (Nonlinear hidden, linear output)

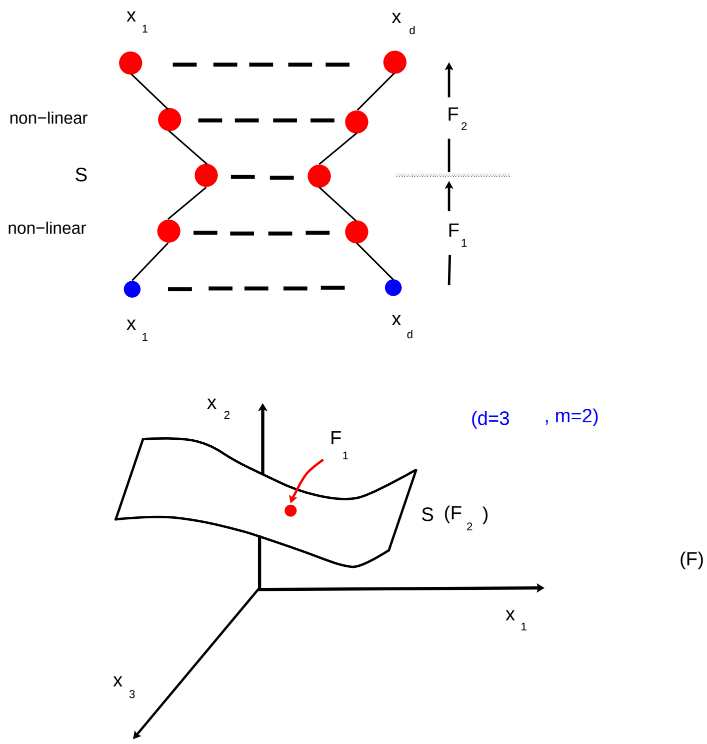
$$\hat{x}_i = \sum_{j=1}^{m} u_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i) \quad \sigma \text{ nonlinear}$$

Objective: Minimize

$$L(W, U) = \sum_{i=1}^{n} \sum_{\ell=1}^{d} (x_i - \hat{x}_i)^2$$

Ineffective Can show that MSE solution is again the principal component subspace

Neural Network PCA II



Autossociative 4 layer network

F_1 Arbitrary mapping possible - universality of neural networks (assumes arbitrary number of first-layer hiddens units)

Optimization Need to solve complex optimization problem - iterative gradient based methods

Local PCA I

Source: Kambhatla and Leen, *Neural Comp.*, 9(7):1493-1516 1997

Motivation: PCA assumes *global linear* structures

Basic idea: 'Everything is *locally* linear'

Method: Quantize domain and apply PCA locally

Local PCA algorithm

1. Partition input space into Q disjoint regions $R^{(i)}$.
2. Compute local covariances
$$\Sigma^{(i)} = \mathbf{E}[(\boldsymbol{x} - \mathbf{E}\boldsymbol{x})(\boldsymbol{x} - \mathbf{E}\boldsymbol{x})^\top | \boldsymbol{x} \in R^{(i)}]$$
and their eigenvectors $\mathbf{e}_j^{(i)}$, $j = 1, \ldots, d$, $\lambda_1^{(i)} \geq \lambda_2^{(i)} \geq \cdots \geq \lambda_d^{(i)} \geq 0$.
3. Choose target dimension m and retain m eigenvectors in each domain.

Local PCA II

Drawback: Partition done prior to coding

Region centroids: $\mathbf{r}^{(i)}$

Local coordinate description:

$$\mathbf{z} = \left(\mathbf{e}_1^{(i)} \cdot (\boldsymbol{x} - \mathbf{r}^{(i)}), \ldots, \mathbf{e}_m^{(i)} \cdot (\boldsymbol{x} - \mathbf{r}^{(i)}) \right) \in \mathbb{R}^m$$

Decoded Vector:

$$\hat{\boldsymbol{x}} = \mathbf{r}^{(i)} + \sum_{j=1}^{m} z_j \mathbf{e}_j^{(i)}$$

Reconstruction error:

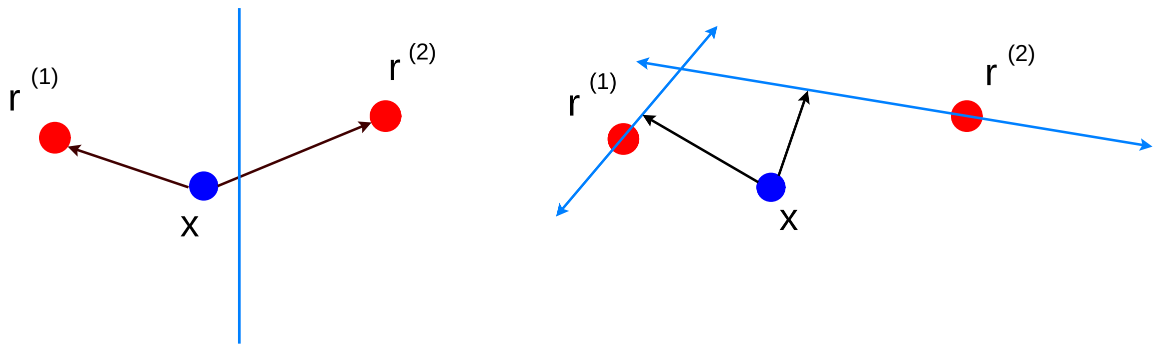
$$\begin{aligned} d(\boldsymbol{x}, r^{(i)}) &= \left\| \boldsymbol{x} - \mathbf{r}^{(i)} - \sum_{j=1}^{m} z_j \mathbf{e}_j^{(i)} \right\|^2 \\ &= (\boldsymbol{x} - \mathbf{r}^{(i)})^\top \Pi^{(i)} (\boldsymbol{x} - \mathbf{r}^{(i)}) \\ \Pi^{(i)} &= I - \Phi_m^{(i)} \Phi_m^{(i)\top} \\ &= \text{projection ortho. to PCA space} \end{aligned}$$

Local PCA III

Idea: Select the center $\mathbf{r}^{(i)}$ so that the distortion is minimized

$$\mathbf{r}^{(i)} = \operatorname*{argmin}_{\mathbf{r}} \left\{ \frac{1}{n_i} \sum_{\boldsymbol{x} \in R^{(i)}} (\boldsymbol{x} - \mathbf{r})^T \Pi^{(i)} (\boldsymbol{x} - \mathbf{r}) \right\}$$

Euclidean and reconstruction distance



Results:

- Excellent results for speech and image coding
- Results far superior to linear approaches and *global* nonlinear methods

Local PCA IV

Idea: Partition using distortion measure

Improved Local PCA algorithm

1. Initialize $\mathbf{r}^{(i)}$ into Q randomly chosen data points; Set $\Sigma^{(i)}$ to the identity
2. Partition data into nearest-neighbor regions $R^{(i)}$ based on $d(\boldsymbol{x}, \mathbf{r}^{(i)})$
3. Recompute centroids based on

$$\mathbf{r}^{(i)} = \underset{\mathbf{r}}{\operatorname{argmin}} \left\{ \frac{1}{n_i} \sum_{\boldsymbol{x} \in R^{(i)}} (\boldsymbol{x} - \mathbf{r})^\top \Pi^{(i)} (\boldsymbol{x} - \mathbf{r}) \right\}$$

4. Recompute variances

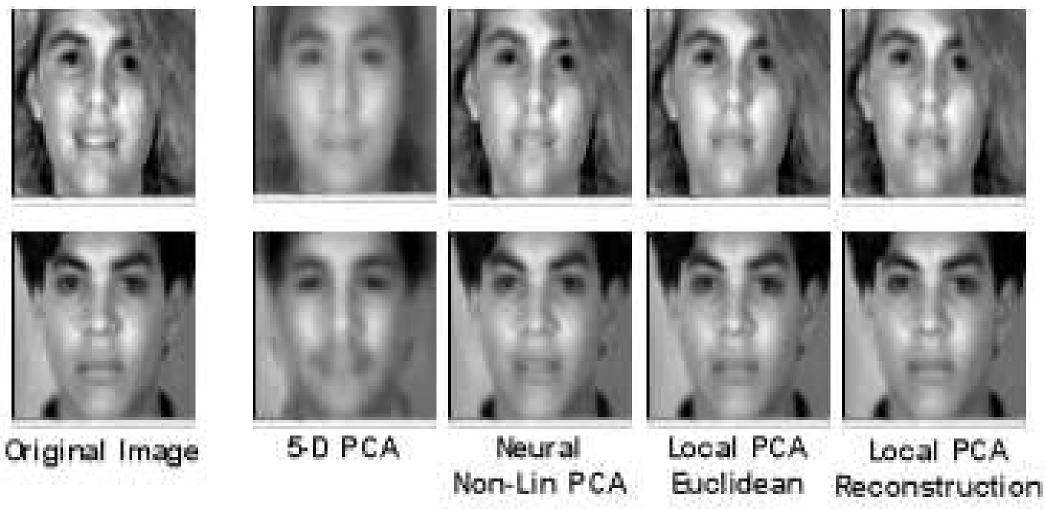
$$\Sigma^{(i)} = \frac{1}{n_i} \sum_{\boldsymbol{x} \in R^{(i)}} (\boldsymbol{x} - \mathbf{r}^{(i)})^\top (\boldsymbol{x} - \mathbf{r}^{(i)})$$

   and use eigenvectors of $\Sigma^{(i)}$ to encode

5. Iterate until reconstruction error falls below a threshold

Local PCA - Image Reduction I

Task:	Compress image database
Input:	160 images of 20 faces
	64×64, 8-bit/pixel grayscale
	Use 4096-dim input vector
Compression:	Use five principal components
Split:	120 train, 20 validation, 20 test



Local PCA - Image Reduction II

Autoassociator: Five layer neural network

Algorithm	**Rec. Error**	**Training (sec.)**
PCA	0.463	5
Autoassoc.	0.327 ± 0.027	4171 ± 41
loc-PCA (Euc.)	0.179 ± 0.048	202 ± 57
loc-PCA (Rec.)	0.173 ± 0.050	62 ± 5

Algorithm	**Enc. Time**	**Dec. Time**
PCA	545	500
Autoassoc.	2750	2750
loc-PCA (Euc.)	3544	500
loc-PCA (Rec.)	91500	500

Local PCA - Image Reduction III

Conclusions:

Reconstruction error: Five-layer network 30% lower than global PCA. Local PCA 40% lower than best auto-associator.

Training time: Local PCA significantly faster than auto-associator.

Encode time: loc-PCA based on Reconstruction distance is slow

Decode time: Either loc-PCA are much faster than auto-associator.

Reproducibility: Because of local minima, auto-associators results vary greatly

Laplacian Eigenmaps I

Source: Laplacian eigenmaps for dimensionality reduction and data representation, Belkin and Niyogi, Neural Computation 15:1373-1396, 2003

The task: Given $\boldsymbol{x}^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathbb{R}^\ell$,

- ⋆ Map $\boldsymbol{x}^n \mapsto \boldsymbol{y}^n = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$
- ⋆ $\boldsymbol{x}_k \in \mathbb{R}^\ell, \ \boldsymbol{y}_k \in \mathbb{R}^m$ and $m \ll \ell$
- ⋆ Underlying assumption: the points $\boldsymbol{x}^n$ lie on a low-dimensional manifold

Algorithmic outline:

- ⋆ Map data onto adjacency graph
- ⋆ Choose weights of the graph
- ⋆ Compute eigenvectors of the Laplacian graph operator

Motivation: Show that this mapping preserves local information optimally (in well defined sense)

Laplacian Eigenmaps - Algorithm

- ⋆ **Construct adjacency graph** Put edge between i and j if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are 'close'. For Example

$$i \leftrightarrow j \quad \text{iff } i \in \text{kNN}(j) \text{ or } j \in \text{kNN}(i)$$

- ⋆ **Choose weights** Using heat kernel

$$W_{ij} = \begin{cases} \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}\right\} & i \text{ and } j \text{ connected} \\ 0 & \text{otherwise} \end{cases}$$

- ⋆ **Eigenmaps** Assume graph connected, otherwise perform for each connected component. Define

$$D_{ii} \triangleq \sum_j W_{ij}, \qquad L \triangleq D - W$$

L is symmetric and positive-semidefinite
Show that

$$2\boldsymbol{x}^\top L\boldsymbol{x} = \sum_{i,j}(x_i - x_j)^2 W_{ij} \geq 0$$

Algorithm - continued

- **Eigenmaps ...** Solve generalized eigenvalue problem

$$Lf = \lambda Df$$

Let $\mathbf{f}_0, \ldots, \mathbf{f}_{n-1}$ be eigenvectors in ascending order of eigenvalues,

$$L\mathbf{f}_k = \lambda_k D\mathbf{f}_k \qquad 0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1}$$

- Construct m-dim mapping (remove $\mathbf{f}_0 = (1, 1, \ldots, 1)$)

$$\boldsymbol{x}_k \mapsto (\mathbf{f}_1(k), \ldots, \mathbf{f}_m(k)) \qquad k\text{-th component}$$

A Note on Optimization

Let L be a non-negative definite matrix.

Consider

$$\begin{aligned} \min \quad & \boldsymbol{y}^\top L \boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{y}^\top D \boldsymbol{y} = 1 \end{aligned}$$

Lagrangian

$$L(\boldsymbol{y}, \lambda) = \boldsymbol{y}^\top L \boldsymbol{y} - \lambda \left(\boldsymbol{y}^\top D \boldsymbol{y} - 1 \right)$$

Setting $\partial L / \partial \boldsymbol{y} = \mathbf{0}$,

$$L\boldsymbol{y} = \lambda D \boldsymbol{y}$$

Generalized eigenvalue problem

Claim The eigenvectors are D-orthogonal.

Proof: Assume $\boldsymbol{x}, \boldsymbol{y}$ eigenvectors with eigenvalues λ, μ, $\lambda \neq \mu$

$$\begin{aligned} \boldsymbol{y}^\top L \boldsymbol{x} &= \boldsymbol{y}^\top \lambda D \boldsymbol{x} = \lambda \boldsymbol{y}^\top D \boldsymbol{x} \\ &= \mu \boldsymbol{y}^\top D \boldsymbol{x} \end{aligned}$$

$$\lambda \neq \mu \quad \Rightarrow \quad \boldsymbol{y}^\top D \boldsymbol{x} = 0$$

Optimal Embedding I

Show that mapping
preserves local information optimally

1D case $\boldsymbol{x}^n \mapsto y^n = \{y_1, \ldots, y_n\}$. Introduce criterion

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = 2\boldsymbol{y}^\top L\boldsymbol{y} \qquad \boldsymbol{y} = (y_1, \ldots, y_n)$$

Remove arbitrary scaling using

$$\boldsymbol{y}^\top D\boldsymbol{y} = 1$$

Optimization problem:

$$\min_{\boldsymbol{y}} \{\boldsymbol{y}^\top L\boldsymbol{y}\} \quad \text{s.t.} \quad \boldsymbol{y}^\top D\boldsymbol{y} = 1$$

Solution: Obtained from generalized eigenvalue problem

$$L\boldsymbol{y} = \lambda D\boldsymbol{y}$$

Optimal Embedding II

Need to solve

$$L\boldsymbol{y} = \lambda D\boldsymbol{y} \qquad (*)$$

Trivial solution $\lambda = 0$ and $\boldsymbol{y} = \mathbf{1}$,

$$L\mathbf{1} = (D - W)\mathbf{1} = \operatorname{diag}\left(\sum_j W_{ij}\right)\mathbf{1} - W\mathbf{1} = \mathbf{0}$$

Eliminate trivial solution $\boldsymbol{y} = \mathbf{1}$ with $\lambda = 0$ by demanding

$$\boldsymbol{y}^\top D\mathbf{1} = 0$$

Obtain new problem

$$\min_{\boldsymbol{y}} \left\{\boldsymbol{y}^\top L\boldsymbol{y}\right\} \quad \text{s.t.} \quad \boldsymbol{y}^\top D\boldsymbol{y} = 1 \quad \& \quad \boldsymbol{y}^\top D\mathbf{1} = 0$$

Solution: Normalized eigenvector of $(*)$ with smallest nonzero eigenvalue

Optimal Embedding III

m-dimensional case Embedding given by

$$Y = [\boldsymbol{y}_1 \boldsymbol{y}_2 \cdots \boldsymbol{y}_m] \in \mathbb{R}^{n \times m}$$
$$\boldsymbol{x}_k \mapsto k\text{th row of } Y$$

Objective: Minimize

$$\min_Y \mathrm{Tr}\left\{Y^\top L Y\right\} \quad \text{s.t.} \quad Y^\top D Y = 1$$

Constraint Prevents collapse into subspace of dimension $m-1$

Solution: Matrix of eigenvectors corresponding to lowest m eigenvalues of

$$L\boldsymbol{y} = \lambda D \boldsymbol{y}$$

Again, remove zero eigenvalue

Optimal Embedding - Figure