# Introduction to Machine Learning and Pattern Recognition

David Brady[1]

[1]ECE Department
Northeastern University

Fall 2018

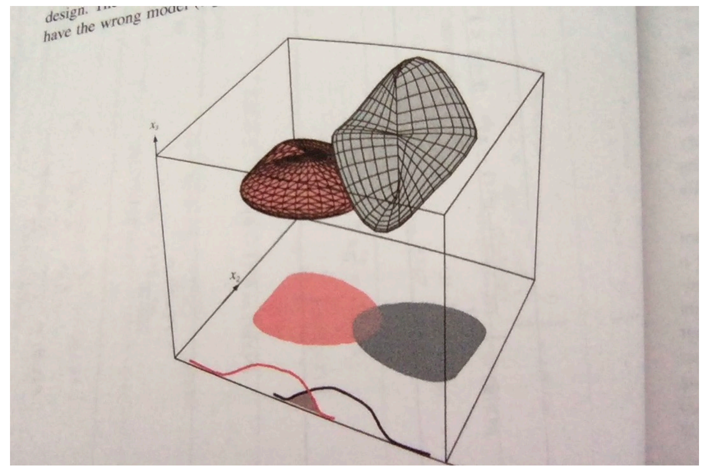
# Next:

- Higher Dimensional Features Are Good
- Lower Dimensional Features Are Good
- Principal Component Analysis
  - Linear PCA
  - Adaptive PCA
  - Nonlinear PCA (after neural net lectures)
  - Kernel PCA
- Fisher Linear Discriminant Analysis
- Kernel LDA
- Multiple Discriminant Analysis
- Independent Component Analysis

# Independent Gaussian Shift in Mean

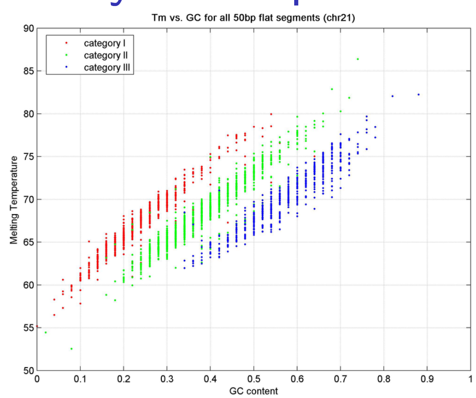
- $p(\mathbf{x}|\omega_i) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}),\ P[\omega_i] = 0.5$
- Bayes Classifier:
  $P[error] = 1/\sqrt{2\pi} \int_{r/2}^{\infty} e^{-u^2/2} du$
- $r = \sqrt{(\boldsymbol{\mu_1} - \boldsymbol{\mu_2})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu_1} - \boldsymbol{\mu_2})}$
  Mahalanobis distance
- Independence
  $\Longrightarrow r^2 = \sum_{i=1}^{d} \frac{\mu_{i1} - \mu_{i2}}{\sigma_i}$ **squared!**

**r grows with d, P[error] drops with d**

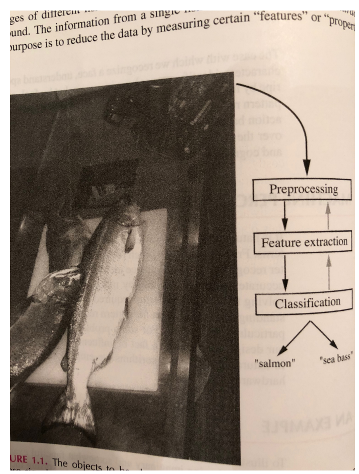


- (non-Gaussian distributions in figure)
- more data is better (?) $\Longrightarrow$ higher $d$

# Reality: Independence Disappears for High Dimensions




- in-class scatter clouds are "flat" for high *d*
- marginal return on feature dimension
- newer dimensions become predictable **(dependent)**



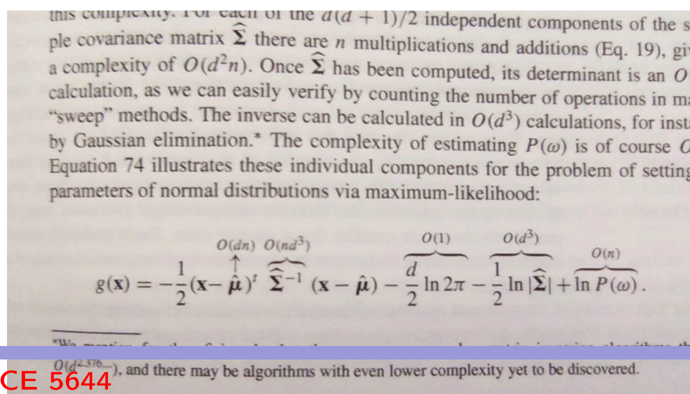

- height, weight, width,length, color, lightness...
- what else provides additional discrimination?

# Next:

- Higher Dimensional Features Are Good
- Lower Dimensional Features Are Good
- Principal Component Analysis
  - Linear PCA
  - Adaptive PCA
  - Nonlinear PCA (after neural net lectures)
  - Kernel PCA
- Fisher Linear Discriminant Analysis
- Kernel LDA
- Multiple Discriminant Analysis
- Independent Component Analysis

# Training, Parameter Estimation, Computational Complexity

- determine Gaussian discriminant for class $i$
- $n$ training feature vectors (fixed $n$ )
- feature dimension $d$ (growing $d$ )



this complexity. For each of the $d(d+1)/2$ independent components of the sa
ple covariance matrix $\widehat{\Sigma}$ there are $n$ multiplications and additions (Eq. 19), giv
a complexity of $O(d^2n)$. Once $\widehat{\Sigma}$ has been computed, its determinant is an $O($
calculation, as we can easily verify by counting the number of operations in ma
"sweep" methods. The inverse can be calculated in $O(d^3)$ calculations, for insta
by Gaussian elimination.* The complexity of estimating $P(\omega)$ is of course $O$
Equation 74 illustrates these individual components for the problem of setting
parameters of normal distributions via maximum-likelihood:

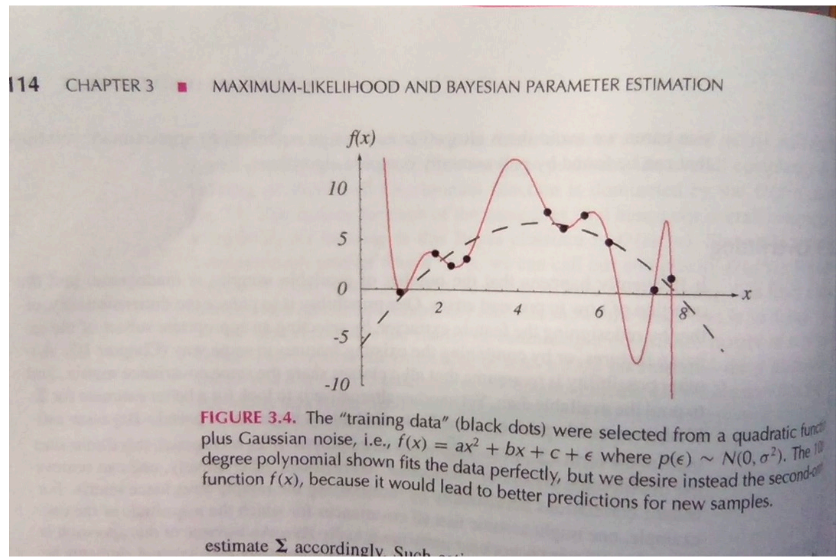
$$g(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\underset{O(dn)}{\hat{\boldsymbol{\mu}}})^t \overbrace{\widehat{\boldsymbol{\Sigma}}^{-1}}^{O(nd^3)} (\mathbf{x}-\hat{\boldsymbol{\mu}}) - \overbrace{\frac{d}{2}\ln 2\pi}^{O(1)} - \overbrace{\frac{1}{2}\ln|\widehat{\boldsymbol{\Sigma}}|}^{O(d^3)} + \overbrace{\ln P(\omega)}^{O(n)}.$$

$O(d^{2.376...})$, and there may be algorithms with even lower complexity yet to be discovered.

- $d + \frac{d(d-1)}{2} \approx \frac{d^2}{2}$ scalar parameters (large $d$ )
- $nd$ scalar training samples
- $2n/d$ samples per parameter $\to 0$ !
- parameter estimator error grows with $d$
- $g(\mathbf{x})$ computational complexity is $\mathscr{O}(nd^3) \to \infty$

# Model Overfitting

- features are corrupted by noise (ex., additive)
- small $d \rightarrow$ model does not follow signal
- large $d \rightarrow$ model follows signal and noise
- Goldilocks $d$ ?




- Information Criteria: AIC, BIC (later)
- Component Analysis (here)

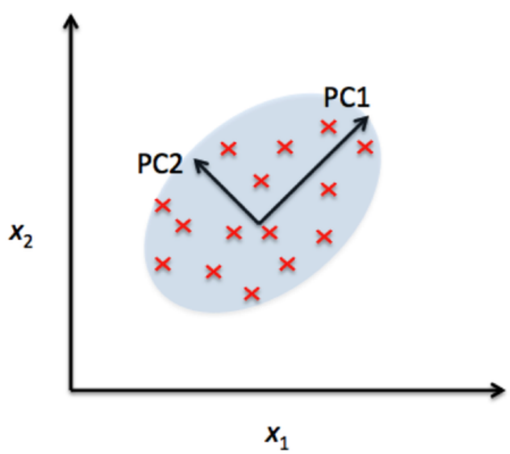
## Next:

- Higher Dimensional Features Are Good
- Lower Dimensional Features Are Good
- Principal Component Analysis
  - Linear PCA
  - Adaptive PCA
  - Nonlinear PCA (after neural net lectures)
  - Kernel PCA
- Fisher Linear Discriminant Analysis
- Kernel LDA
- Multiple Discriminant Analysis
- Independent Component Analysis

# Linear PCA

- sample $\mathbf{x_1}, \ldots \mathbf{x_n}$. Find best representative vector $\mathbf{x_0}$.
- $\mathbf{x_0} = \arg\min_{\mu} \sum_{k=1}^{n} \|\mu - \mathbf{x}_k\|^2 \rightarrow$ $\mathbf{x_0} = \mathbf{m} = \sum \mathbf{x_k}/n$
- sample mean best represents data
- but is there something better?
- let $\mathbf{x}_k \approx \mathbf{m} + a_k \mathbf{e}$. Find $\{a_k\}, \mathbf{e}$!
- $\{\mathbf{a}, \ \mathbf{e}\} = \arg\min_{\|\mathbf{e}\|=1} \sum_{j=1}^{n} \|\mathbf{x}_k - a_k \mathbf{e} - \mathbf{m}\|^2 \Longrightarrow$ $a_k = \mathbf{e}^t (\mathbf{x_k} - \mathbf{m})$
- scatter matrix $\mathbf{S} = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t$
- substitution yields: $\mathbf{e} = \arg\min_{\|\mathbf{v}\|=\mathbf{1}} -\mathbf{v}^t \mathbf{S} \mathbf{v} + \sum_{k=1}^{n} \|\mathbf{x_k} - \mathbf{m}\|^2$
- $\mathbf{e} = \arg\max_{\|\mathbf{v}\|=\mathbf{1}} \mathbf{v}^t \mathbf{S} \mathbf{v} \rightarrow$ Appendix A.3 $\rightarrow \mathbf{S}\mathbf{e} = \lambda \mathbf{e}$
- $\mathbf{e}$ is the dominant eigenvector of $\mathbf{S}$ (principal component)
- $a_k$ is the projection of $\mathbf{x_k} - \mathbf{m}$ onto $\mathbf{e}$
- $\lambda = \mathbf{e^t S e}$ is the principal value

# Linear PCA (2)



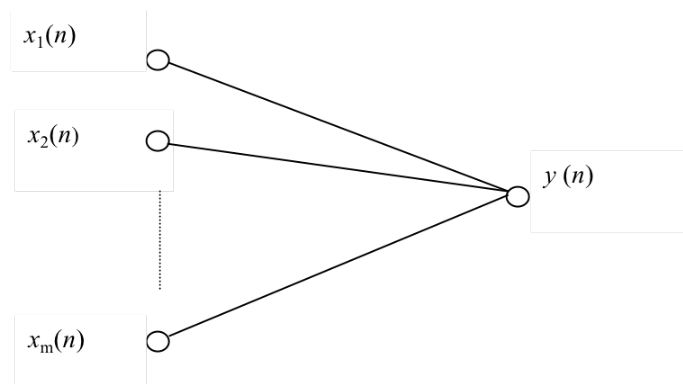

- 2 dimensions here, 2 principal components (no reduction)
- also, $[\mathbf{E}, \mathbf{D}] = eig(\mathbf{S})$

Recursively find other principal components:

- initialize $\mathbf{S}_1 = \mathbf{S},\ \mathbf{e}_1 = \mathbf{e},\ \lambda_1 = \lambda$
- $\mathbf{S}_{i+1} = \mathbf{S_i} - \lambda_i \mathbf{e_i}\mathbf{e_i^t}$
- Linear $\mathrm{PCA}(\mathbf{S}_{i+1}) \rightarrow \lambda_{i+1}, \mathbf{e_{i+1}}$
- stop when $\lambda_1/\lambda_{K+1} - \lambda_1/\lambda_K < \varepsilon$
- $\{\lambda_i, \mathbf{e_i}\}_{i=1}^{K}$ are the principal values, components

# On-line Version of PCA

- left neurons, activation $\mathbf{x}(n) \in \mathscr{R}^m$
- right neuron connected by weights $\mathbf{w}(n) \in \mathscr{R}^m$
- right neuron activation $y(n) = \mathbf{w}^t(n)\mathbf{x}(n) \in \mathscr{R}$
- find sequence $\{\mathbf{w}(n)\} \rightarrow \mathbf{w}_{opt} = \mathbf{e}$




- let $\mathbf{S}(n) = \mathbf{x}(n)\mathbf{x}^t(n)$ be a single-step estimate of $\mathbf{S}$

# Adaptive PCA (2)

- $J_0(\mathbf{v}) = \frac{\mathbf{vS(n)v}}{\|\mathbf{v}\|^2}$
- $y^2(n) = \mathbf{v}^t(n)\mathbf{S}(n)\mathbf{v}(n)$
- $J_0(\mathbf{v}) = \frac{y^2(n)}{\|\mathbf{v}\|^2}$
- $\mathbf{w}_{opt} = \arg max_{\mathbf{v(n)}} J_0(\mathbf{v}(n))$
- consider the stochastic ascent:
- $\mathbf{w}(n+1) - \mathbf{w}(n) = \eta \mathbf{D}(n)$, where
- $\mathbf{D}(n)$ approximates $\nabla J_0$

two steps:

1. $\hat{\mathbf{w}}(n+1) = \mathbf{w}(n) + \eta y(n)\mathbf{w}(n)$
2. $\mathbf{w}(n+1) = \hat{\mathbf{w}}(n+1) / \|\hat{\mathbf{w}}(n+1)\|$

- for small $\eta$, single-step approximation

1. $\mathbf{w}(n+1) - \mathbf{w}(n) = \eta\left(\mathbf{x}(n) - y(n)\mathbf{w}(n)\right) y(n) + \mathscr{O}(\eta^2)$

# Kernel PCA

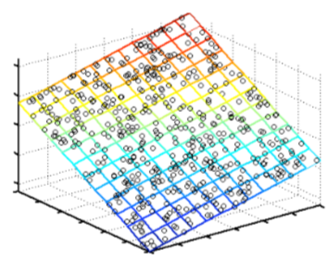
## Dimensionality Reduction

- Data representation

  Inputs are real-valued vectors in a high dimensional space.

- Linear structure

  Does the data live in a low dimensional subspace?



- Nonlinear structure

  Does the data live on a low dimensional submanifold?

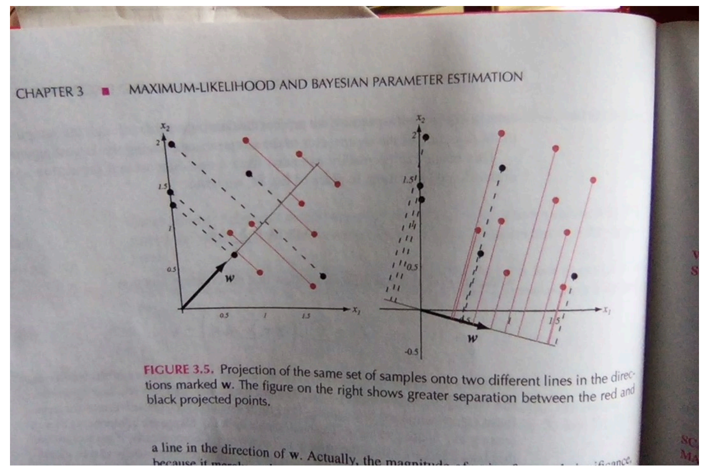
# KPCA hyperlink

[L08]KPCA

## Next:

# LDA

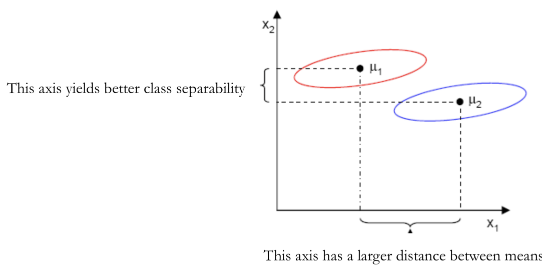
- 2-classes $\omega_1, \ \omega_2$
- observe $\mathbf{x}_i, \ i = 1 \ldots n.$
- form $y_i = \mathbf{w}^t \mathbf{x}_i$ to separate classes
- let $\mathbf{m}_i = \sum_{j \in \mathscr{D}_i} \mathbf{x}_j / n_i$
- class scatter matrix $\mathbf{S}_i = \sum_{\mathbf{x} \in \mathscr{D}_\mathbf{i}} (\mathbf{x} - \mathbf{m_i})(\mathbf{x} - \mathbf{m_i})^t$
- want $|\mathbf{w}^t (\mathbf{m}_1 - \mathbf{m}_2)|$ large $\rightarrow \|\mathbf{w}\| = \infty$




- better: $J(\mathbf{w}) = |\mathbf{w}^t (\mathbf{m}_1 - \mathbf{m}_2)|^2 / \mathbf{w}^t (\mathbf{S}_1 + \mathbf{S}_2) \mathbf{w}$ large

# LDA (2)

- $J(\mathbf{w}) = \mathbf{w}^t \mathbf{S}_B \mathbf{w} / \mathbf{w}^t \mathbf{S}_W \mathbf{w}$ (generalized Rayleigh quotient)
- $\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$ (between-class scatter matrix)
- $\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$ (within-class scatter matrix)
- calculus of variations: $\delta / \delta \varepsilon \;\; J(\mathbf{w}_o + \varepsilon \mathbf{v})|_{\varepsilon=0} = 0 \forall \mathbf{v}$

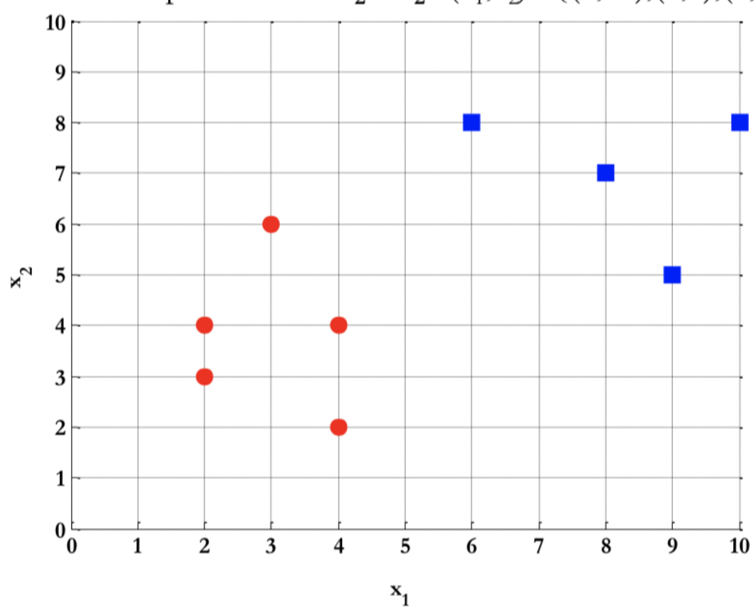



- generalized eigenvector: $\mathbf{S}_B \mathbf{w_o} = \lambda \mathbf{S}_W \mathbf{w_o}$
- $\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w}_o = \lambda \mathbf{w}_o$, if $\mathbf{S}_W^{-1}$exists
- $\mathbf{w}_o = \mathbf{S}_W^{-1} (\mathbf{m_1} - \mathbf{m_2})$, since $\mathbf{m}_1 - \mathbf{m}_2 \;\alpha\; \mathbf{S}_B^{-1} \mathbf{w}_o$
- $\mathbf{w}_o$ is canonical variate

# LDA Example

## LDA … Two Classes - Example

- Compute the Linear Discriminant projection for the following two-dimensional dataset.
  - Samples for class $\omega_1$ : $\mathbf{X}_1=(x_1,x_2)=\{(4,2),(2,4),(2,3),(3,6),(4,4)\}$
  - Sample for class $\omega_2$ : $\mathbf{X}_2=(x_1,x_2)=\{(9,10),(6,8),(9,5),(8,7),(10,8)\}$




```
% samples for class 1
X1 = [4,2;
      2,4;
      2,3;
      3,6;
      4,4];


% samples for class 2
X2 = [9,10;
      6,8;
      9,5;
      8,7;
      10,8];
```

# LDA hyperlink

[L09]Elhabian_LDA09

# Next:

# Next:

# MDA

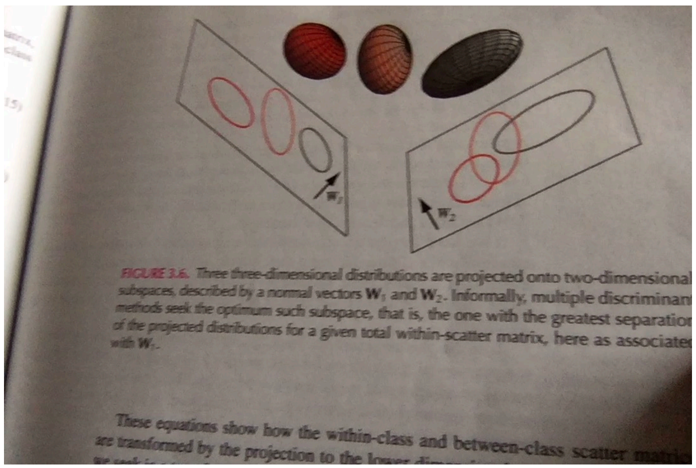


FIGURE 3.6. Three three-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors $W_1$ and $W_2$. Informally, multiple discriminant methods seek the optimum such subspace, that is, the one with the greatest separation of the projected distributions for a given total within-scatter matrix, here as associated with $W_1$.

These equations show how the within-class and between-class scatter matrices are transformed by the projection to the lower dim

- c classes & c-1 discriminants
- $n_i$ features in class $\omega_i$
- total mean $\mathbf{m} = \sum_{i=1}^{c} \mathbf{m}_i n_i / n$
- total scatter matrix $\mathbf{S}_T = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t$
- $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$ , $\mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t$
- we seek $\mathbf{W} \in \mathscr{R}^{d \times (c-1)}$ to yield $\mathbf{y} = \mathbf{W^t x}$
- good $\mathbf{W} \iff$ large between-class scatter, small within-class scatter
- $J(\mathbf{W}) = \left|\mathbf{W^t S_B W}\right| / \left|\mathbf{W^t S_W W}\right|$
- solution-> ith column of $\mathbf{W}$: $\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i$
- implementation: roots of char poly $|\mathbf{S}_B - \lambda_i \mathbf{S}_W| = 0$
- then solve: $(\mathbf{S}_B - \lambda_i \mathbf{S}_W)\mathbf{w}_i = \mathbf{0}$
- $\mathbf{S}_B$ has rank $\leq c-1 \implies$ at most $c-1$ positive eigenvalues

# MDA

## LDA … C-Classes

- Now, we have *C*-classes instead of just two.
- We are now seeking (*C-1*) projections [$\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{C-1}$] by means of (*C-1*) projection vectors $\mathbf{w}_i$.
- $\mathbf{w}_i$ can be arranged by *columns* into a projection matrix $\mathbf{W} = [\mathbf{w}_1 | \mathbf{w}_2 | \ldots | \mathbf{w}_{C-1}]$ such that:

$$y_i = w_i^T x \quad \Rightarrow \quad y = W^T x$$

$$where \quad x_{m\times 1} = \begin{bmatrix} x_1 \\ . \\ . \\ x_m \end{bmatrix} \quad , \quad y_{C-1\times 1} = \begin{bmatrix} y_1 \\ . \\ . \\ y_{C-1} \end{bmatrix}$$

$$and \quad W_{m\times C-1} = \left[ w_1 | \quad w_2 | \quad \ldots \quad | w_{C-1} \right]$$

# MDA hyperlink

Elhabian_LDA09.pdf

## Next:

- Higher Dimensional Features Are Good
- Lower Dimensional Features Are Good
- Principal Component Analysis
  - Linear PCA
  - Adaptive PCA
  - Nonlinear PCA (after neural net lectures)
  - Kernel PCA
- Fisher Linear Discriminant Analysis
- Kernel LDA
- Multiple Discriminant Analysis
- Independent Component Analysis

# 2-Column Template