INTRODUCTORY APPLIED MACHINE LEARNING

Yan-Fu Kuo

Dept. of Bio-industrial Mechatronics Engineering

National Taiwan University

Today:

- Linear discriminant analysis
- General discriminant analysis

Outline

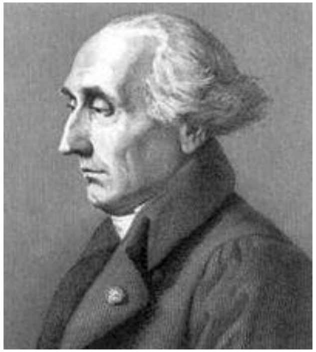
- Goal of the lecture
- Math review – Lagrange multiplier
- Linear discriminant analysis
- General discriminant analysis

Goals

- After this, you should be able to:
  - Understand basic principals of discriminant analysis
  - Perform discriminant analysis
  - Be able to determine what type of discriminant analysis to be carried out

History of Lagrange Multiplier

- Named after Joseph Louis Lagrange



- A strategy for finding the maxima/minima of a function subject to constraints
- Provides a <u>necessary condition</u> for optimality in constrained problems

Lagrange Multiplier

- Consider an optimization problem

  Minimize $f(x, y)$

  subject to $g(x, y) = c$

- Lagrangian:

$$L(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c),$$

  where $\lambda \in \Re$ is the Lagrange multiplier

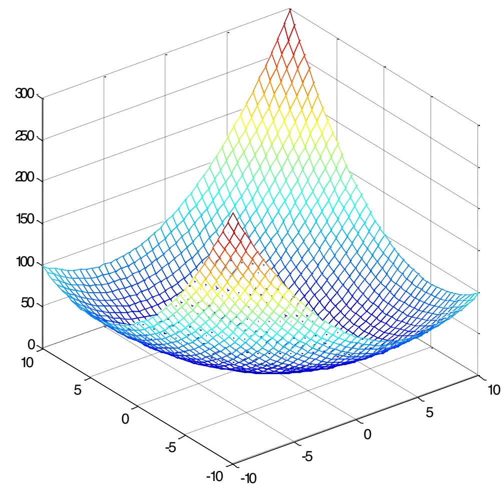
- Let (x^*, y^*) be a local minimizer of $f(\cdot)$ subject to $g(\cdot)$, then there exists λ such that the partial derivatives of $L(x, y, \lambda)$ are zero

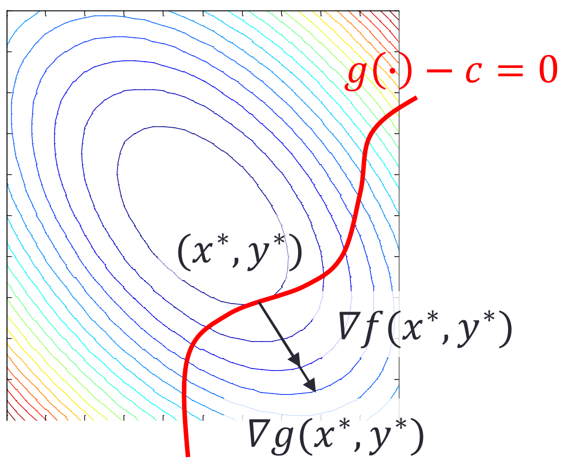
Geometric Explanation

- Example function:

$$f(x, y) = x^2 + xy + y^2$$

- The value of $f(\cdot)$ can vary while moving along the contour line for $g(\cdot) = c$
- Only when the contour line for $g(\cdot) = c$ meets contour lines of $f(\cdot)$ tangentially, the value of $f(\cdot)$ does not increase or decrease
- Hence a local minimum or maximum





Geometric Explanation Matlab Code

```
% plot quadratic function and contour lines
[x, y] = meshgrid(-10:.5:10,-10:.5:10);
z = x.^2 + x.*y + y.^2; % x^2 + x*y + y^2
mesh( x, y, z);
xlim([-10 10]); ylim([-10 10]);
xlabel('x_1'); ylabel('x_2'); zlabel('f(x_1,x_2)');
set( gcf, 'Color', 'w')

figure;
[C,h] = contour( x, y, z, 20); set( gcf, 'Color', 'w')
xlim([-10 10]); ylim([-10 10]); xlabel('x_1');
ylabel('x_2');
```

Lagrange Multiplier (Cont'd)

- At the local minimum or maximum(x^*, y^*),

$$\nabla f(x^*, y^*) = \lambda \nabla g(x^*, y^*)$$

- To incorporate these conditions into one equation, we introduce an auxiliary function

$$L(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c),$$

and solve

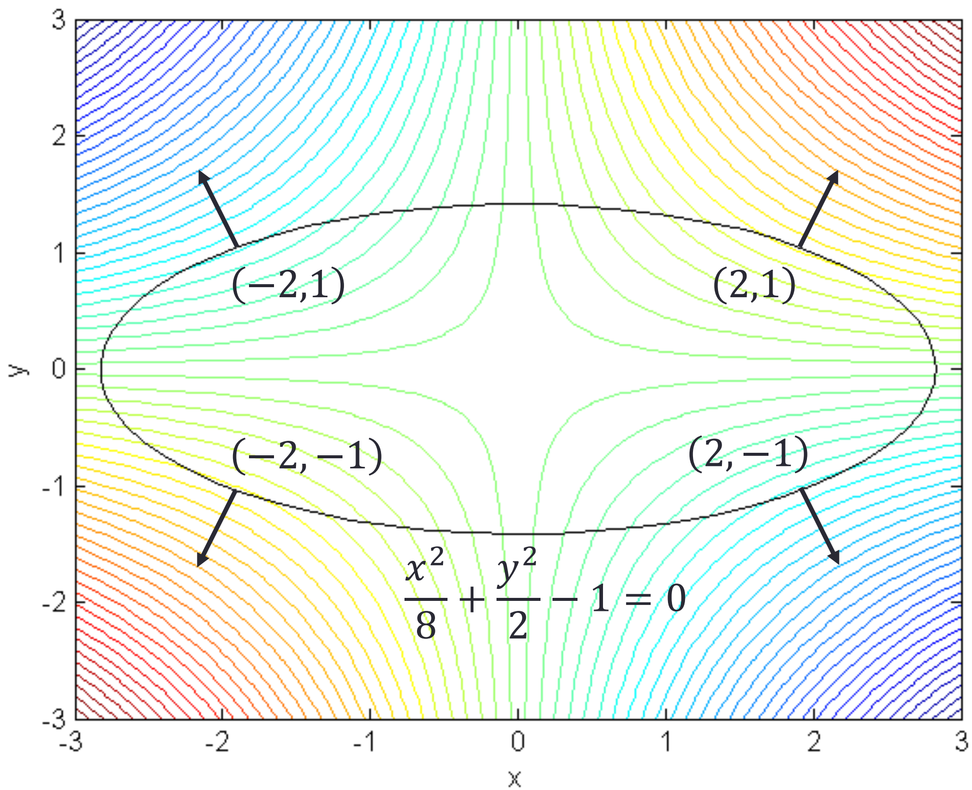
$$\nabla L(x, y, \lambda) = \mathbf{0}$$

Example

- Function $f(x, y) = xy$

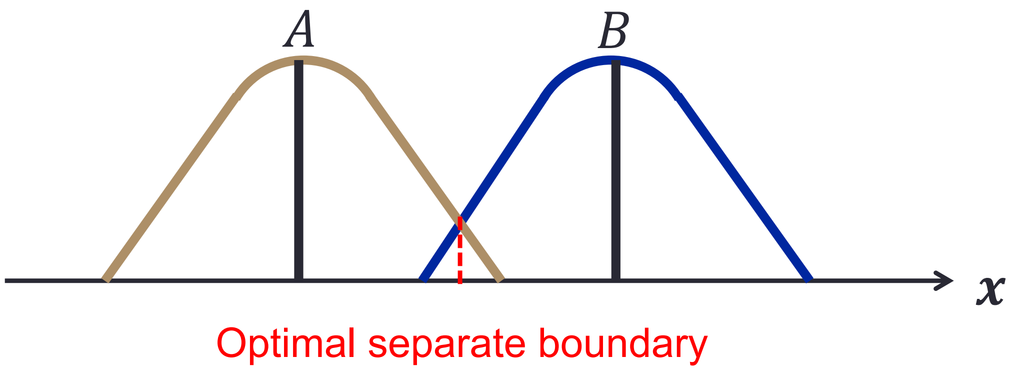
  subject to $g(x, y) = \frac{x^2}{8} + \frac{y^2}{2} - 1$
- Lagrangian: $L(x, y, \lambda) = xy - \lambda\left(\frac{x^2}{8} + \frac{y^2}{2} - 1\right)$
- Gradient of Lagrangian: $\nabla L(x, y, \lambda) = \begin{pmatrix} y - \frac{\lambda x}{4} \\ x - \lambda y \\ \frac{x^2}{8} + \frac{y^2}{2} - 1 \end{pmatrix} = \mathbf{0}$

Geometric Explanation of the Example



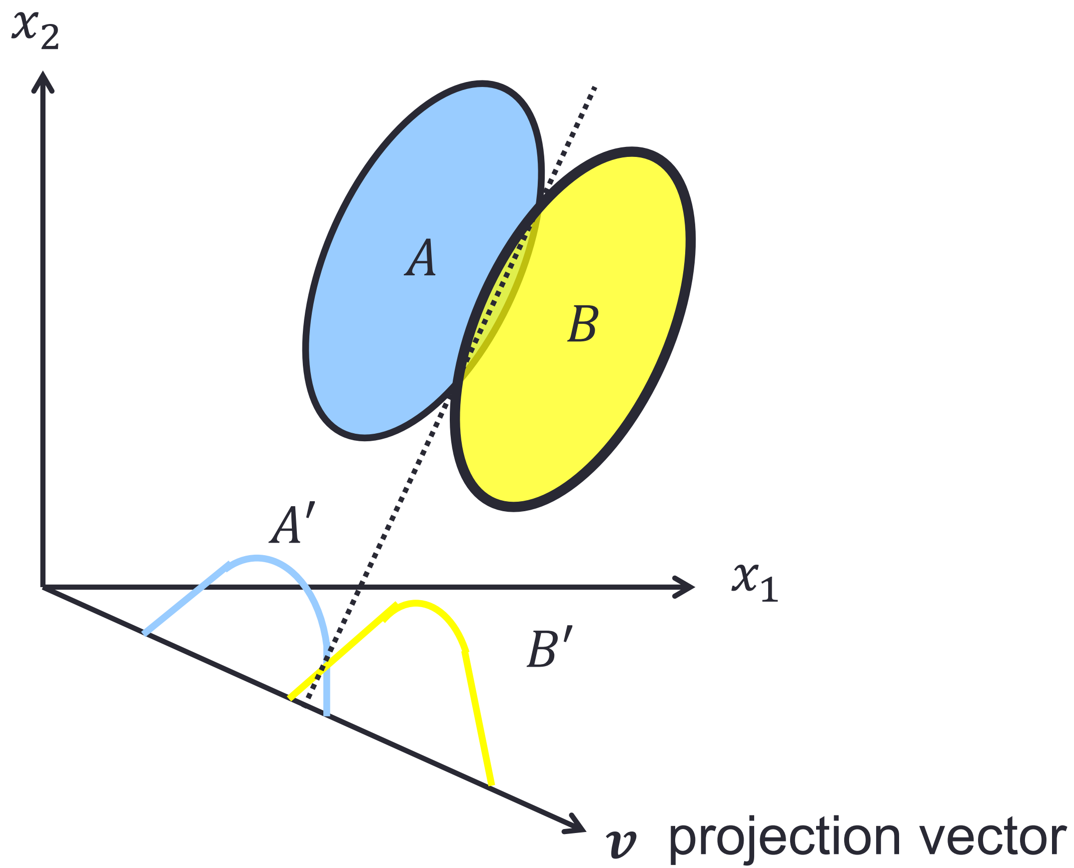
Discriminant Analysis

- The objective is to identify boundaries between groups of objects, i.e., classification
- Example: univariate discriminant analysis:



- Usually applied on high-dimensional data
- Perform dimensionality reduction while preserving as much of the class discriminatory information as possible

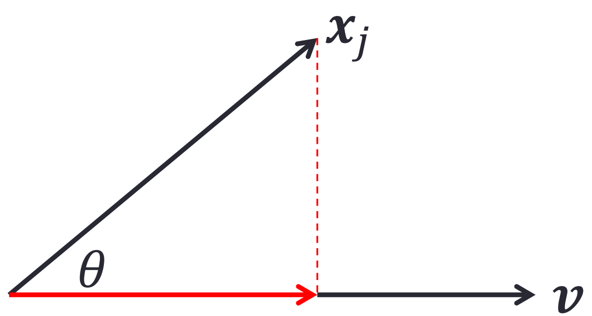
Illustration of Two-group Discriminant Analysis



Linear Discriminant Analysis (LDA)

- Originally developed in 1936 by R. A. Fisher
- Split the total scatter into within-classes scatter as well as the between-classes scatter (brought from the idea of ANOVA)
- In LDA, the objective is to find a projection vector $\boldsymbol{v}$ such that:
  1. The distance of projections of class means is the largest
  2. The distance between projections of samples in every class and the projection of corresponding class mean is the smallest

Recall: Vector Projection



$$\left(\|\boldsymbol{x}_j\|\cos\theta\right)\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} = \|\boldsymbol{x}_j\|\frac{\boldsymbol{x}_j{}^{\mathrm{T}}\boldsymbol{v}}{\|\boldsymbol{x}_j\|\|\boldsymbol{v}\|}\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} = \frac{\boldsymbol{x}_j{}^{\mathrm{T}}\boldsymbol{v}}{\|\boldsymbol{v}\|^2}\boldsymbol{v}$$

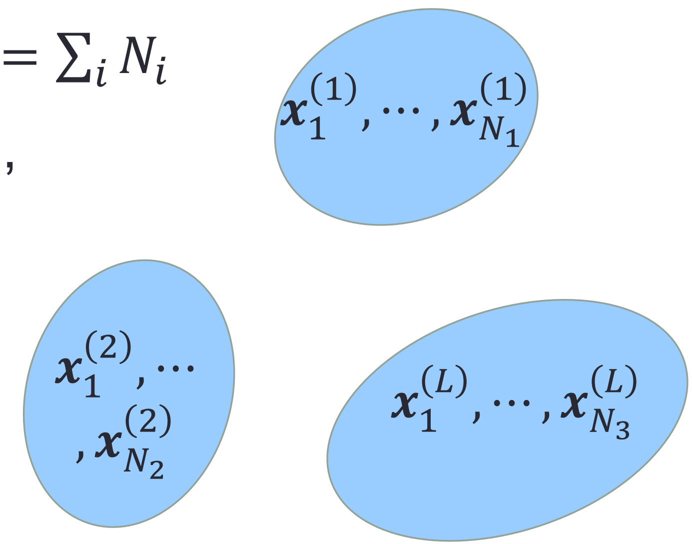
If $\|\boldsymbol{v}\| = 1$, then $(\|\boldsymbol{x}_j\|\cos\theta)\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} = (\boldsymbol{x}_j{}^{\mathrm{T}}\boldsymbol{v})\boldsymbol{v}$

Notations

- $\boldsymbol{x}_j^{(i)} \in \Re^d$: the jth sample in class i,

  where $j = 1 \dots N_i$ and $i = 1 \dots L$
- N_i: number of samples in class i
- L: number of classes
- N: number of all samples, i.e., $N = \sum_i N_i$
- $\boldsymbol{m}_i \in \Re^d$: the mean of class i, i.e.,

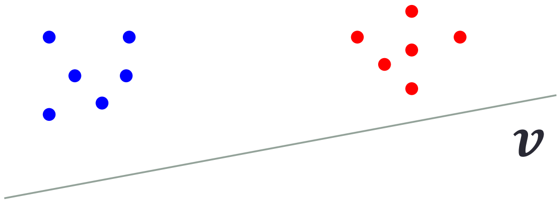
$$\boldsymbol{m}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} \boldsymbol{x}_j^{(i)}$$



Objective and Strategy

- Objective:

  Find a vector $\boldsymbol{v}$ such that the projected distance of the data points between different classes on $\boldsymbol{v}$ are maximized



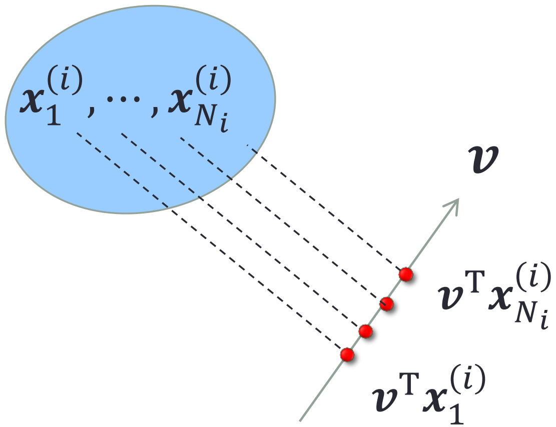
- Strategy:

  1. Define the between-class scatter matrix $\boldsymbol{S}_b^{LDA} \in \Re^{d \times d}$ and within-class scatter matrix $\boldsymbol{S}_w^{LDA} \in \Re^{d \times d}$
  2. Find $\boldsymbol{v}$ with which the between-class variance $\boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_b^{LDA} \boldsymbol{v}$ is maximized while the within-class variance $\boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_w^{LDA} \boldsymbol{v}$ is minimized

Mean of Projected Data Points

- For a given fector $\boldsymbol{v} \in \Re^d$, the projections of all the points $\boldsymbol{x}_j^{(i)}$ onto it are

$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_1^{(1)}, \cdots, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_{N_1}^{(1)},$$
$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_1^{(2)}, \cdots, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_{N_2}^{(2)},$$
$$\cdots$$
$$\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_1^{(L)}, \cdots, \boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_{N_L}^{(L)}$$

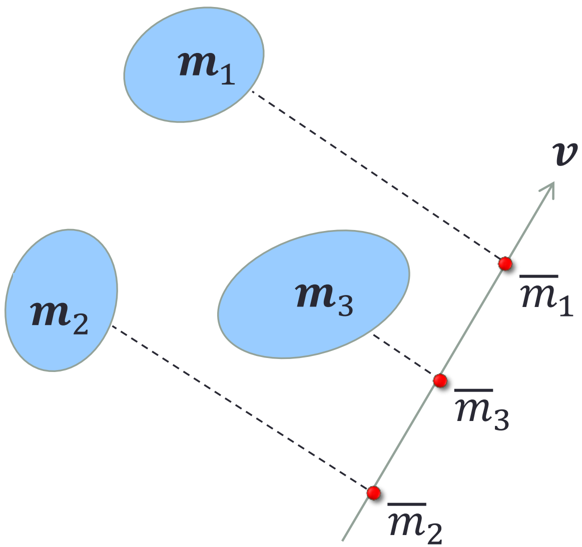


- The mean of the projected data points of class i is

$$\overline{m}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} \boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_j^{(i)} = \boldsymbol{v}^{\mathrm{T}}\left(\frac{1}{N_i}\sum_{j=1}^{N_i} \boldsymbol{x}_j^{(i)}\right) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{m}_i$$

Between-class Scatter

- Define the projected sum of squared between-class variance:



$$\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\frac{N_i}{N}\frac{N_j}{N}(\overline{m}_i-\overline{m}_j)^2 \in \Re$$

$$=\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\frac{N_i}{N}\frac{N_j}{N}(\overline{m}_i-\overline{m}_j)(\overline{m}_i-\overline{m}_j)^{\mathrm{T}}$$

$$=\sum_{i=1}^{L-1}\sum_{j=i+1}^{L}\frac{N_i}{N}\frac{N_j}{N}(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{m}_i-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{m}_j)(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{m}_i-\boldsymbol{v}^{\mathrm{T}}\boldsymbol{m}_j)^{\mathrm{T}}$$

Between-class Scatter

$$= \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \frac{N_i}{N} \frac{N_j}{N} \boldsymbol{v}^{\mathrm{T}} (\boldsymbol{m}_i - \boldsymbol{m}_j)(\boldsymbol{m}_i - \boldsymbol{m}_j)^{\mathrm{T}} \boldsymbol{v}$$

$$= \boldsymbol{v}^{\mathrm{T}} \left(\sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \frac{N_i}{N} \frac{N_j}{N} (\boldsymbol{m}_i - \boldsymbol{m}_j)(\boldsymbol{m}_i - \boldsymbol{m}_j)^{\mathrm{T}} \right) \boldsymbol{v}$$

$$= \boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_b^{LDA} \boldsymbol{v} \in \Re$$

- Define $\boldsymbol{S}_b^{LDA} \in \Re^{d \times d}$ as between-class scatter matrix, which is independent of $\boldsymbol{v}$
- $\boldsymbol{S}_b^{LDA}$ is a symmetric positive-definite matrix and is of rank $d - 1$ or less

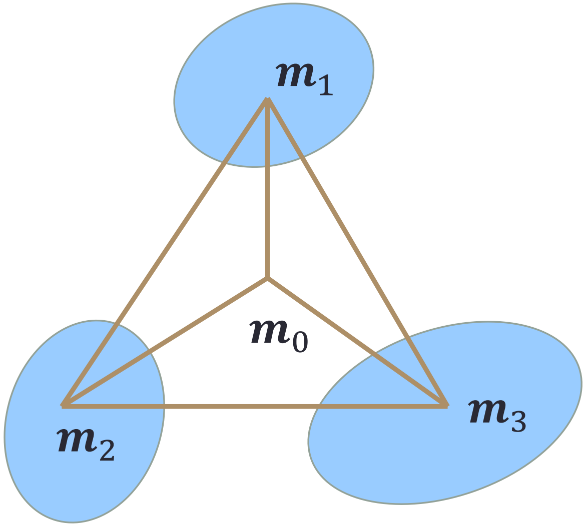
Geometric Interpretation of $\boldsymbol{S}_b^{LDA}$

- The between-class scatter matrix:

$$\boldsymbol{S}_b^{LDA} = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \frac{N_i}{N} \frac{N_j}{N} (\boldsymbol{m}_i - \boldsymbol{m}_j)(\boldsymbol{m}_i - \boldsymbol{m}_j)^{\mathrm{T}}$$

$$= \sum_{i=1}^{L} \frac{N_i}{N} (\boldsymbol{m}_i - \boldsymbol{m}_0)(\boldsymbol{m}_i - \boldsymbol{m}_0)^{\mathrm{T}}$$

- Define $\boldsymbol{m}_0 \equiv \sum_{i=1}^{L} \frac{N_i}{N} \boldsymbol{m}_i$



Between-class Scatter Matrix $\boldsymbol{S}_b^{LDA}$

$$\boldsymbol{S}_b^{LDA} = \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \frac{N_i}{N} \frac{N_j}{N} (\boldsymbol{m}_i - \boldsymbol{m}_j)(\boldsymbol{m}_i - \boldsymbol{m}_j)^{\mathrm{T}}$$

$$= \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{N_i}{N} \frac{N_j}{N} (\boldsymbol{m}_i - \boldsymbol{m}_j)(\boldsymbol{m}_i - \boldsymbol{m}_j)^{\mathrm{T}}$$

$$= \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{N_i}{N} \frac{N_j}{N} (\boldsymbol{m}_i \boldsymbol{m}_i^{\mathrm{T}} - \boldsymbol{m}_i \boldsymbol{m}_j^{\mathrm{T}} - \boldsymbol{m}_j \boldsymbol{m}_i^{\mathrm{T}} + \boldsymbol{m}_j \boldsymbol{m}_j^{\mathrm{T}})$$

$$= \frac{1}{2} \left(\sum_{i=1}^{L} \sum_{j=1}^{L} \frac{N_i}{N} \frac{N_j}{N} \boldsymbol{m}_i \boldsymbol{m}_i^{\mathrm{T}} - \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{N_i}{N} \frac{N_j}{N} \boldsymbol{m}_i \boldsymbol{m}_j^{\mathrm{T}} \right.$$

Between-class Scatter Matrix $\boldsymbol{S}_b^{LDA}$ (Cont'd)

$$\boldsymbol{S}_b^{LDA} = \frac{1}{2}\left(\sum_{j=1}^{L}\frac{N_j}{N}\sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}} - \sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\sum_{j=1}^{L}\frac{N_j}{N}\boldsymbol{m}_j^{\mathrm{T}}\right.$$

$$\left. - \sum_{j=1}^{L}\frac{N_j}{N}\boldsymbol{m}_j\sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i^{\mathrm{T}} + \sum_{i=1}^{L}\frac{N_i}{N}\sum_{j=1}^{L}\frac{N_j}{N}\boldsymbol{m}_j\boldsymbol{m}_j^{\mathrm{T}}\right)$$

- Define $\boldsymbol{m}_0 \equiv \sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i$

Between-class Scatter Matrix $\boldsymbol{S}_b^{LDA}$ (Cont'd)

$$\boldsymbol{S}_b^{LDA} = \frac{1}{2}\left(\sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}} - \boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}} - \boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}} + \sum_{j=1}^{L}\frac{N_j}{N}\boldsymbol{m}_j\boldsymbol{m}_j^{\mathrm{T}}\right)$$

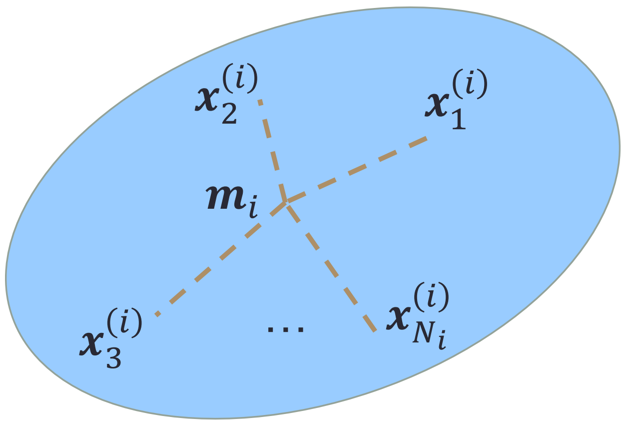
$$= \sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}} - \boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}} - \boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}} + \boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}}$$

$$= \sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}} - \sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\,\boldsymbol{m}_0^{\mathrm{T}} - \boldsymbol{m}_0\left(\sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_i\right)^{\mathrm{T}} + \sum_{i=1}^{L}\frac{N_i}{N}\boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}}$$

$$= \sum_{i=1}^{L}\frac{N_i}{N}\left[\boldsymbol{m}_i\boldsymbol{m}_i^{\mathrm{T}} - \boldsymbol{m}_i\boldsymbol{m}_0^{\mathrm{T}} - \boldsymbol{m}_0\boldsymbol{m}_i^{\mathrm{T}} + \boldsymbol{m}_0\boldsymbol{m}_0^{\mathrm{T}}\right]$$

$$= \sum_{i=1}^{L}\frac{N_i}{N}(\boldsymbol{m}_i - \boldsymbol{m}_0)(\boldsymbol{m}_i - \boldsymbol{m}_0)^{\mathrm{T}}$$

Within-class Scatter Matrix S_W^{LDA}



Within-class Scatter Matrix S_W^{LDA}

- Define the projected sum of squared within-class variance:

$$\sum_{i=1}^{L}\sum_{j=1}^{N_i}\frac{1}{N_i}(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_j^{(i)}-\overline{m}_i)(\boldsymbol{v}^{\mathrm{T}}\boldsymbol{x}_j^{(i)}-\overline{m}_i)^{\mathrm{T}}\in\Re$$

$$=\sum_{i=1}^{L}\sum_{j=1}^{N_i}\frac{1}{N_i}\boldsymbol{v}^{\mathrm{T}}(\boldsymbol{x}_j^{(i)}-\boldsymbol{m}_i)(\boldsymbol{x}_j^{(i)}-\boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{v}$$

$$=\boldsymbol{v}^{\mathrm{T}}\left(\sum_{i=1}^{L}\sum_{j=1}^{N_i}\frac{1}{N_i}(\boldsymbol{x}_j^{(i)}-\boldsymbol{m}_i)(\boldsymbol{x}_j^{(i)}-\boldsymbol{m}_i)^{\mathrm{T}}\right)\boldsymbol{v}=\boldsymbol{v}^{\mathrm{T}}\boldsymbol{S}_W^{LDA}\boldsymbol{v}$$

- Define $\boldsymbol{S}_W^{LDA}\in\Re^{d\times d}$ as within-class scatter matrix, which is symmetric positive-semidefinite

LDA Formulation

- The optimal projection vector $\boldsymbol{v}$ can be found by the following equation:

$$\boldsymbol{v} = \underset{v \in \Re^d}{\arg\max} \frac{\boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_b^{LDA} \boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_w^{LDA} \boldsymbol{v}} = \underset{\boldsymbol{v}^T \boldsymbol{S}_w^{LDA} \boldsymbol{v} = 1}{\arg\max} \boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_b^{LDA} \boldsymbol{v}$$

or equivalently in Lagrange form:

$$f(\boldsymbol{v}, \lambda) = \boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_b^{LDA} \boldsymbol{v} - \lambda(\boldsymbol{v}^{\mathrm{T}} \boldsymbol{S}_w^{LDA} \boldsymbol{v} - 1)$$

Solving LDA Problem

- Lagrangian:

$$\frac{\partial f}{\partial \boldsymbol{v}} = 2\boldsymbol{S}_b^{LDA}\boldsymbol{v} - 2\lambda\boldsymbol{S}_w^{LDA}\boldsymbol{v} = 0$$

$$\Rightarrow \boldsymbol{S}_b^{LDA}\boldsymbol{v} = \lambda\boldsymbol{S}_w^{LDA}\boldsymbol{v}$$

- This is a generalized eigenvalue problem
- Since $\boldsymbol{S}_b^{LDA}$ is symmetric positive-definite, it can be written as

$$\boldsymbol{S}_b^{LDA} = (\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}$$

where $(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}$ is constructed from eigenvalue decomposition, i.e., $\boldsymbol{S}_b^{LDA} = U\Lambda U^{\mathrm{T}}$ and $(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^{\mathrm{T}}$

Solving LDA Problem (Cont'd)

- Defining $\boldsymbol{w} = (\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}\boldsymbol{v}$, one get

$$(\boldsymbol{S}_w^{LDA})^{-1}(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}\boldsymbol{v} = \lambda\boldsymbol{v}$$

$$\Rightarrow (\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}(\boldsymbol{S}_w^{LDA})^{-1}(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}\boldsymbol{w} = \lambda(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}\boldsymbol{v}$$

$$\Rightarrow (\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}(\boldsymbol{S}_w^{LDA})^{-1}(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}\boldsymbol{w} = \lambda\boldsymbol{w} \quad \ldots\ldots\ldots\ldots\ldots(*)$$

which is a regular eigenvalue problem for a symmetric, positive definite matrix $(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}(\boldsymbol{S}_w^{LDA})^{-1}(\boldsymbol{S}_b^{LDA})^{\frac{1}{2}}$

- Find solution of $\boldsymbol{w}$ from (*) and one can get $\boldsymbol{v}$ from this relationship: $\boldsymbol{v} = (\boldsymbol{S}_b^{LDA})^{\frac{-1}{2}}\boldsymbol{w}$

Optimal Project Vector of Two-class LDA

- Suppose there are only two classes, i.e., $L = 2$
- The optimal projection vector $\boldsymbol{v}$ is

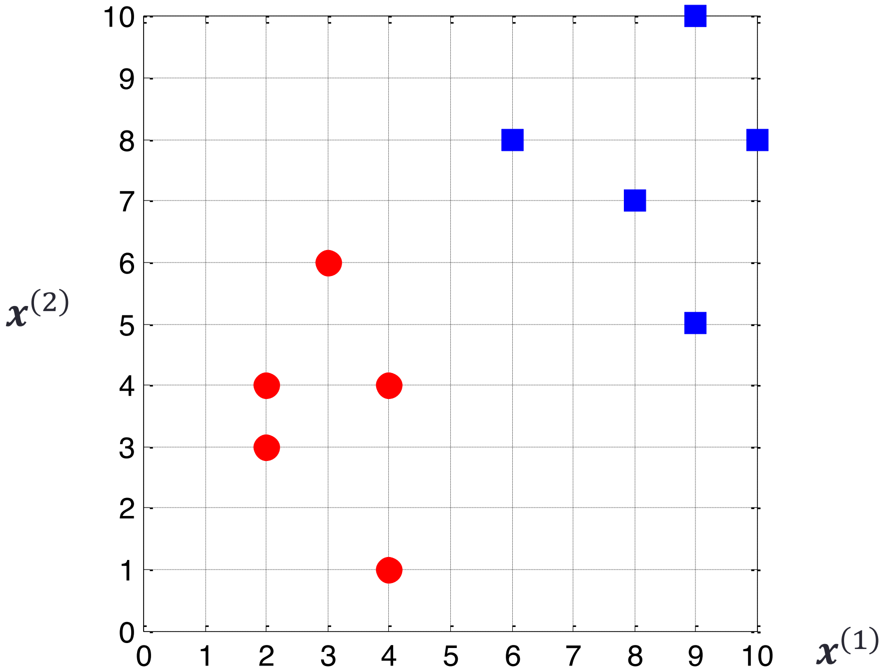
$$\boldsymbol{v} = (\boldsymbol{S}_W^{LDA})^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2) \quad \text{..................(@)}$$

Example

- Compute the LDA projection for the following 2D dataset

$$\boldsymbol{x}^{(1)} = \{(4,1), (2,4), (2,3), (3,6), (4,4)\}$$

$$\boldsymbol{x}^{(2)} = \{(9,10), (6,8), (9,5), (8,7), (10,8)\}$$



Example Solution

- The class means, $\boldsymbol{S}_b^{LDA}$, and $\boldsymbol{S}_w^{LDA}$ are

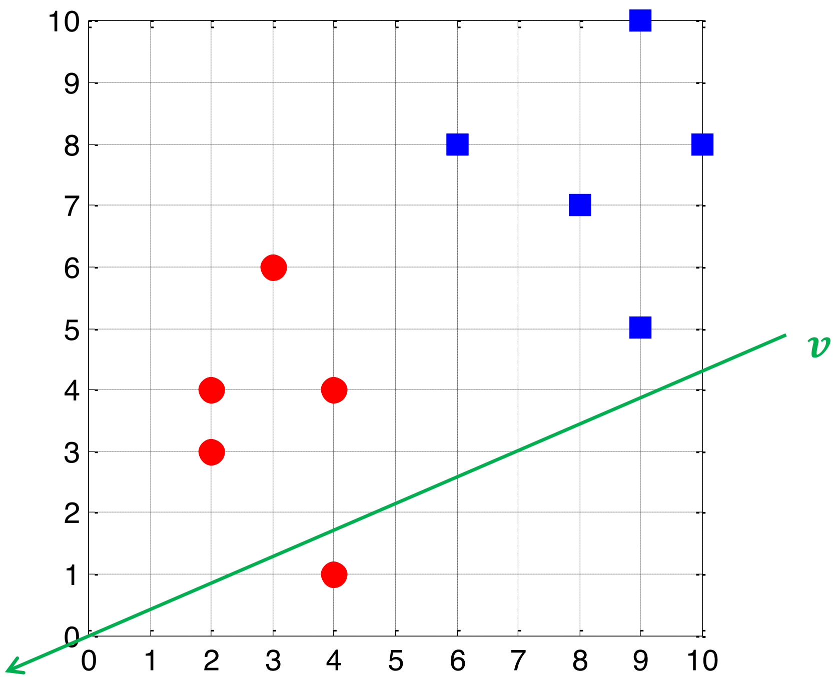
$$\boldsymbol{m}_1 = \begin{bmatrix} 3.0 \\ 3.6 \end{bmatrix}, \qquad \boldsymbol{m}_2 = \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix}$$

$$\boldsymbol{S}_b^{LDA} = \begin{bmatrix} 29.16 & 21.6 \\ 21.6 & 16.0 \end{bmatrix}, \qquad \boldsymbol{S}_w^{LDA} = \begin{bmatrix} 2.64 & -.44 \\ -.44 & 5.28 \end{bmatrix}$$

- Directly by (@), the optimal projection vector $\boldsymbol{v}$ is

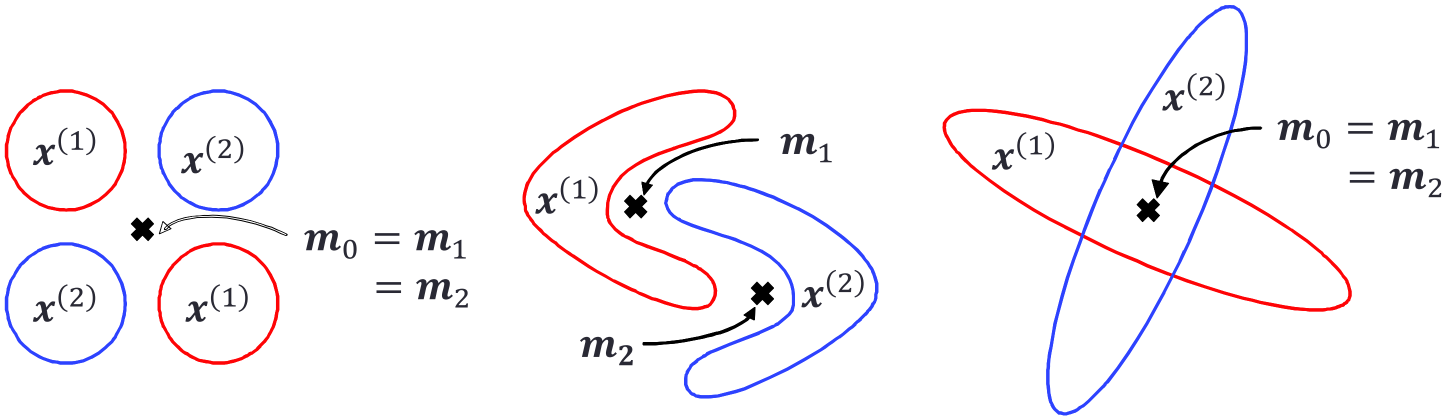
$$\boldsymbol{v} = \left(\begin{bmatrix} 2.64 & -.44 \\ -.44 & 5.28 \end{bmatrix}\right)^{-1} \left(\begin{bmatrix} 3.0 \\ 3.6 \end{bmatrix} - \begin{bmatrix} 8.4 \\ 7.6 \end{bmatrix}\right) = \begin{bmatrix} -.91 \\ -.39 \end{bmatrix}$$

The Optimal Projection Vector $\boldsymbol{v}$



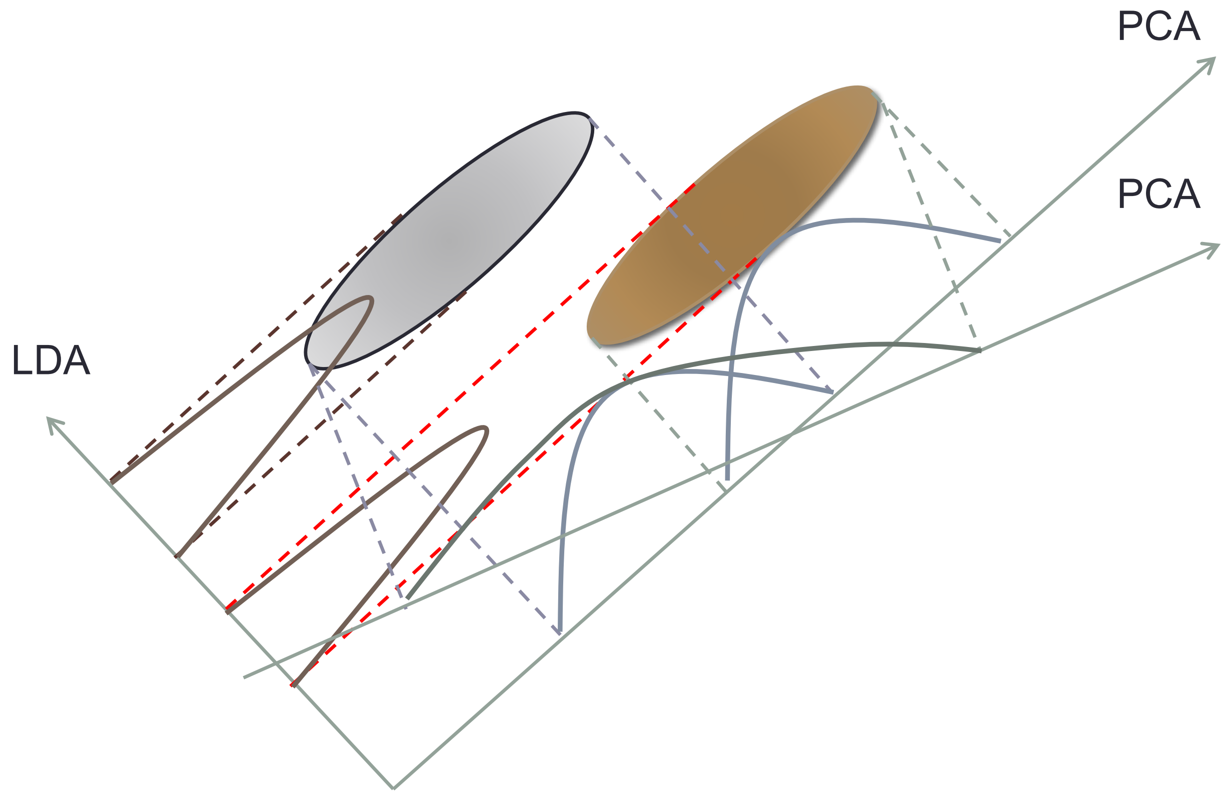
Limitation of LDA

- LDA produces at most $L-1$ feature projections
- LDA is a parametric method (such that it assumes the data points are in Gaussian distribution)

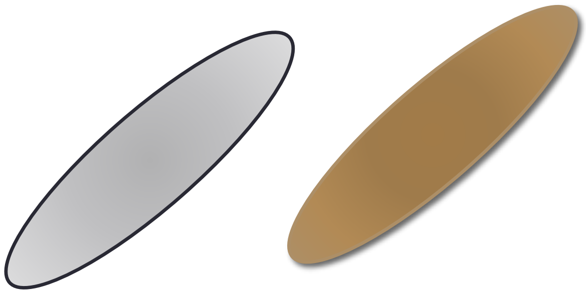


- LDA also fails if discriminatory information is not in the mean but in the variance of the data

LDA vs. PCA



LDA vs. PCA



Generalized Discriminant Analysis (GDA)

- What if the separation of the data points with LDA is not good?
- One solution is to apply kernel methods to the LDA problem – called generalized discriminant analysis (GDA)
- Suppose kernel function $\phi(\cdot): \Re^d \ni \boldsymbol{x}_j^{(i)} \rightarrow \phi(\boldsymbol{x}_j^{(i)}) \in \Re^p$ is applied
- Perform LDA on $\phi(\boldsymbol{x}_j^{(i)})$ instead
- Remember, we only know $< \phi\left(\boldsymbol{x}_j^{(i)}\right), \phi\left(\boldsymbol{x}_j^{(i)}\right) >$, not $\phi(\boldsymbol{x}_j^{(i)})$

Notations

- L: number of classes
- N_i: number of samples in class i
- N: number of all samples, i.e., $N = \sum_i N_i$
- $\phi(\boldsymbol{x}_j^{(i)}) \in \Re^p$: the jth sample in class i
- $\boldsymbol{X}_i^{\mathrm{T}} = \left[\phi(\boldsymbol{x}_1^{(i)}), \ldots, \phi(\boldsymbol{x}_{N_i}^{(i)})\right]$
- $\boldsymbol{X}^{\mathrm{T}} = \left[\boldsymbol{X}_1^{\mathrm{T}}, \ldots, \boldsymbol{X}_L^{\mathrm{T}}\right]$

Within- and Between- class Scatter Matrices

- Suppose that the samples in the $\mathcal{H}$ space are centered, i.e.,

$$\boldsymbol{m}_0 = 0$$

- The within-class scatter matrix:

$$\boldsymbol{S}_w^{GDA} = \sum_{i=1}^{L} \sum_{j=1}^{N_i} \frac{1}{N} \phi(\boldsymbol{x}_j^{(i)}) \phi(\boldsymbol{x}_j^{(i)})^{\mathrm{T}}$$

- The between-class scatter matrix:

$$\boldsymbol{S}_b^{GDA} = \sum_{i=1}^{L} \frac{N_i}{N} (\boldsymbol{m}_i - \boldsymbol{m}_0)(\boldsymbol{m}_i - \boldsymbol{m}_0)^{\mathrm{T}} = \sum_{i=1}^{L} \frac{N_i}{N} \boldsymbol{m}_i \boldsymbol{m}_i{}^{\mathrm{T}}$$

Between-class Scatter Matrix

- From the definition

$$\boldsymbol{m}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\phi(\boldsymbol{x}_j^{(i)}) = \frac{1}{N_i}\left[\phi(\boldsymbol{x}_1^{(i)}),\ldots,\phi(\boldsymbol{x}_{N_i}^{(i)})\right]\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}_{N_i\times 1}$$

$$= \frac{1}{N_i}\boldsymbol{X}_i^{\mathrm{T}}\mathbf{1}_{N_i\times 1}$$

And

$$\boldsymbol{m}_i{\boldsymbol{m}_i}^T = \frac{1}{{N_i}^2}\boldsymbol{X}_i^{\mathrm{T}}\mathbf{1}_{N_i\times 1}\mathbf{1}_{1\times N_i}\boldsymbol{X}_i = \frac{1}{{N_i}^2}\boldsymbol{X}_i^{\mathrm{T}}\mathbf{1}_{N_i\times N_i}\boldsymbol{X}_i = \frac{1}{N_i}\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{B}_i\boldsymbol{X}_i$$

where

$$\boldsymbol{B}_i = \frac{1}{N_i}\mathbf{1}_{N_i\times N_i}$$

Between-class Scatter Matrix (Cont'd)

$$\boldsymbol{S}_b^{GDA} = \sum_{i=1}^{L} \frac{N_i}{N} \boldsymbol{m}_i \boldsymbol{m}_i^{\mathrm{T}} = \frac{1}{N} \sum_{i=1}^{L} N_i \frac{1}{N_i} \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{B}_i \boldsymbol{X}_i = \frac{1}{N} \sum_{i=1}^{L} \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{B}_i \boldsymbol{X}_i$$

$$= \frac{1}{N} [\boldsymbol{X}_1^T, \ldots, \boldsymbol{X}_L^T] \begin{bmatrix} \boldsymbol{B}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{B}_L \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_L \end{bmatrix} = \frac{1}{N} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{B} \boldsymbol{X}$$

where $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{B}_L \end{bmatrix}$

Within-class Scatter Matrix

$$\boldsymbol{S}_{W}^{GDA} = \sum_{i=1}^{L} \sum_{j=1}^{N_i} \frac{1}{N} \phi(\boldsymbol{x}_j^{(i)}) \phi(\boldsymbol{x}_j^{(i)})^{\mathrm{T}}$$

$$= \frac{1}{N} \sum_{i=1}^{L} \left[\phi(\boldsymbol{x}_1^{(i)}), \dots, \phi\left(\boldsymbol{x}_{N_i}^{(i)}\right) \right] \begin{bmatrix} \phi\left(\boldsymbol{x}_1^{(i)}\right) \\ \vdots \\ \phi\left(\boldsymbol{x}_{N_i}^{(i)}\right) \end{bmatrix}$$

$$= \frac{1}{N} \sum_{i=1}^{L} \boldsymbol{X}_i^{\mathrm{T}} \boldsymbol{X}_i = \frac{1}{N} \left[\boldsymbol{X}_1^{\mathrm{T}}, \dots, \boldsymbol{X}_L^{\mathrm{T}} \right] \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_L \end{bmatrix} = \frac{1}{N} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{X}$$

GDA Formulation

- The optimal projection vector $\boldsymbol{v}$ can be found by the following equation:

$$\boldsymbol{S}_b^{GDA}\boldsymbol{v} = \lambda \boldsymbol{S}_w^{GDA}\boldsymbol{v}$$

i.e.,

$$(\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{X})\boldsymbol{v} = \lambda(\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{v}$$

where we know $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ but not $\boldsymbol{X}$

Solving GDA Problem

- Suppose that $\boldsymbol{v}$ is a linear combination of all training samples, i.e.,

$$\boldsymbol{v} = \sum_{i=1}^{L} \sum_{j=1}^{N_i} \alpha_j^{(i)} \phi(\boldsymbol{x}_j^{(i)}) = \boldsymbol{X}^{\mathrm{T}} \boldsymbol{\alpha} \qquad \text{where } \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1^{(1)} \\ \vdots \\ \alpha_{N_1}^{(1)} \\ \alpha_1^{(2)} \\ \vdots \\ \alpha_{N_2}^{(1)} \\ \vdots \\ \alpha_1^{(L)} \\ \vdots \\ \alpha_{N_L}^{(1)} \end{bmatrix}_{N\times 1}$$

Solving GDA Problem (Cont'd)

- The GDA problem:

$$\boldsymbol{S}_b^{GDA}\boldsymbol{v} = \lambda \boldsymbol{S}_w^{GDA}\boldsymbol{v}$$

$$(\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{X})\boldsymbol{v} = \lambda(\frac{1}{N}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{v}$$

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha} = \lambda\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}$$

$$\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha} = \lambda\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha}$$

- Let $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$, the problem can be re-written as:

$$(\boldsymbol{K}\boldsymbol{B}\boldsymbol{K})\boldsymbol{\alpha} = \lambda(\boldsymbol{K}\boldsymbol{K})\boldsymbol{\alpha}$$

- Note we only obtain $\boldsymbol{\alpha}$, not $\boldsymbol{v}$ explicitly

GDA Classifier

- To classify an unknown sample point $\boldsymbol{x}$, the following formulation is applied:

$$\boldsymbol{v}^{\mathrm{T}}\phi(\boldsymbol{x}) = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\alpha})^{\mathrm{T}}\phi(\boldsymbol{x}) = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{X}\phi(\boldsymbol{x})$$

$$= \boldsymbol{\alpha}^{\mathrm{T}}\begin{bmatrix} \phi\left(\boldsymbol{x}_1^{(i)}\right)^{\mathrm{T}} \\ \vdots \\ \phi\left(\boldsymbol{x}_{N_i}^{(L)}\right)^{\mathrm{T}} \end{bmatrix}\phi(\boldsymbol{x})$$

$$= \boldsymbol{\alpha}^{\mathrm{T}}\begin{bmatrix} < \phi\left(\boldsymbol{x}_1^{(i)}\right), \phi(\boldsymbol{x}) > \\ \vdots \\ < \phi\left(\boldsymbol{x}_{N_i}^{(L)}\right), \phi(\boldsymbol{x}) > \end{bmatrix}$$

Summary

- LDA and GDA reduce dimension of data while preserving as much of the class discriminatory information as possible
- Kernel methods are applied on problems that cannot be solved with LDA

References

- G. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*