Discrimination and Classification

Nathaniel E. Helwig

Assistant Professor of Psychology and Statistics
University of Minnesota (Twin Cities)

Updated 14-Mar-2017

Copyright

Outline of Notes

1) Classifying Two Populations
- Overview of Problem
- Cost of Misclassification

2) Two Multivariate Normals
- Equal Covariance
- Unequal Covariance

3) Evaluating Classifications
- Misclassification Measures
- Quality in LDA

4) Classifying $g \geq 2$ Populations
- Overview of Problem
- Cost of Misclassification
- Discriminant Analysis

5) Iris Data Example
- Data Overview
- LDA Example
- QDA Example

Purpose of Discrimination and Classification

Discrimination attempts to separate distinct sets of objects, and classification attempts to allocate new objects to predefined groups.

There are two typical goals of discrimination and classification:

1. Data description: find "discriminants" that best separate groups
2. Data allocation: put new objects in groups via the "discriminants"

Note that goal 1 is discrimination, and goal 2 is classification/allocation.

Classifying Two Populations

The Two Population Classification Problem

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ denote a random vector and let

- $f_1(\mathbf{x})$ denote the probability density function (pdf) for population π_1
- $f_2(\mathbf{x})$ denote the probability density function (pdf) for population π_2

Problem: Given a realization $\boldsymbol{X} = \mathbf{x}$, we want to assign $\mathbf{x}$ to π_1 or π_2.

We want to find some classification rule to determine whether a realization $\boldsymbol{X} = \mathbf{x}$ should be assigned to population π_1 or π_2.

Visualizing a Classification Rule

Let Ω denote the sample space, i.e., all possible values of $\mathbf{x}$, and

- $R_1 \subset \Omega$ is the subset of Ω for which we classify $\mathbf{x}$ as π_1
- $R_2 = \Omega - R_1$ is the subset of Ω for which we classify $\mathbf{x}$ as π_2

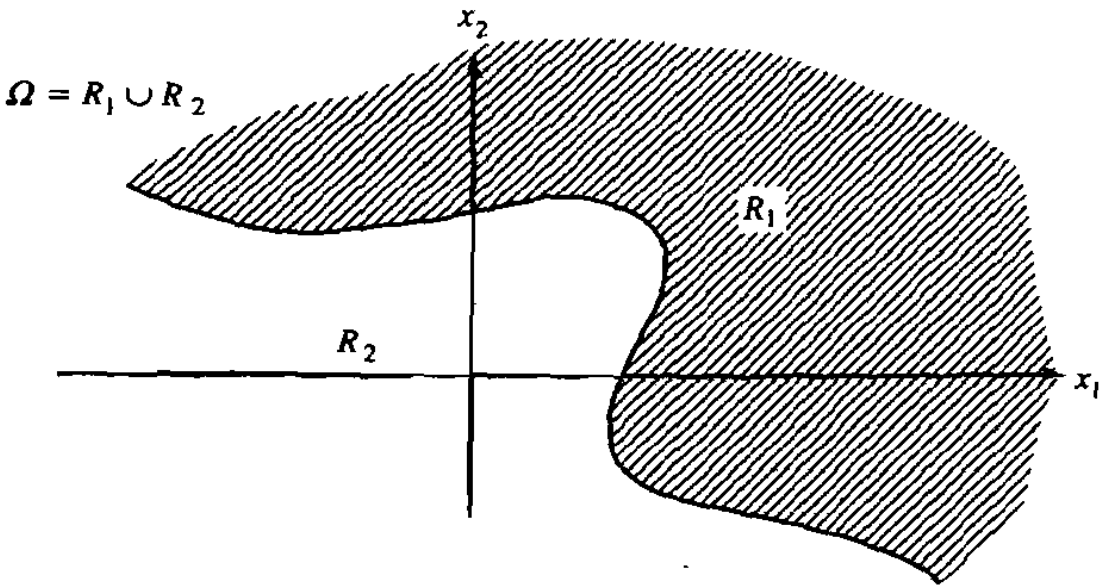


Figure 11.2 Classification regions for two populations.

Figure: Figure 11.2 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern). Visualization is for $p = 2$ variables.

Probability of Misclassification

The conditional probability $P(2|1)$ of classifying an object as π_2 when the object really belongs to π_1 is given by

$$P(2|1) = P(\boldsymbol{X} \in R_2|\pi_1) = \int_{R_2} f_1(\mathbf{x})\mathrm{d}\mathbf{x}$$

The conditional probability $P(1|2)$ of classifying an object as π_1 when the object really belongs to π_2 is given by

$$P(1|2) = P(\boldsymbol{X} \in R_1|\pi_2) = \int_{R_1} f_2(\mathbf{x})\mathrm{d}\mathbf{x}$$

Visualizing the Probability of Misclassification

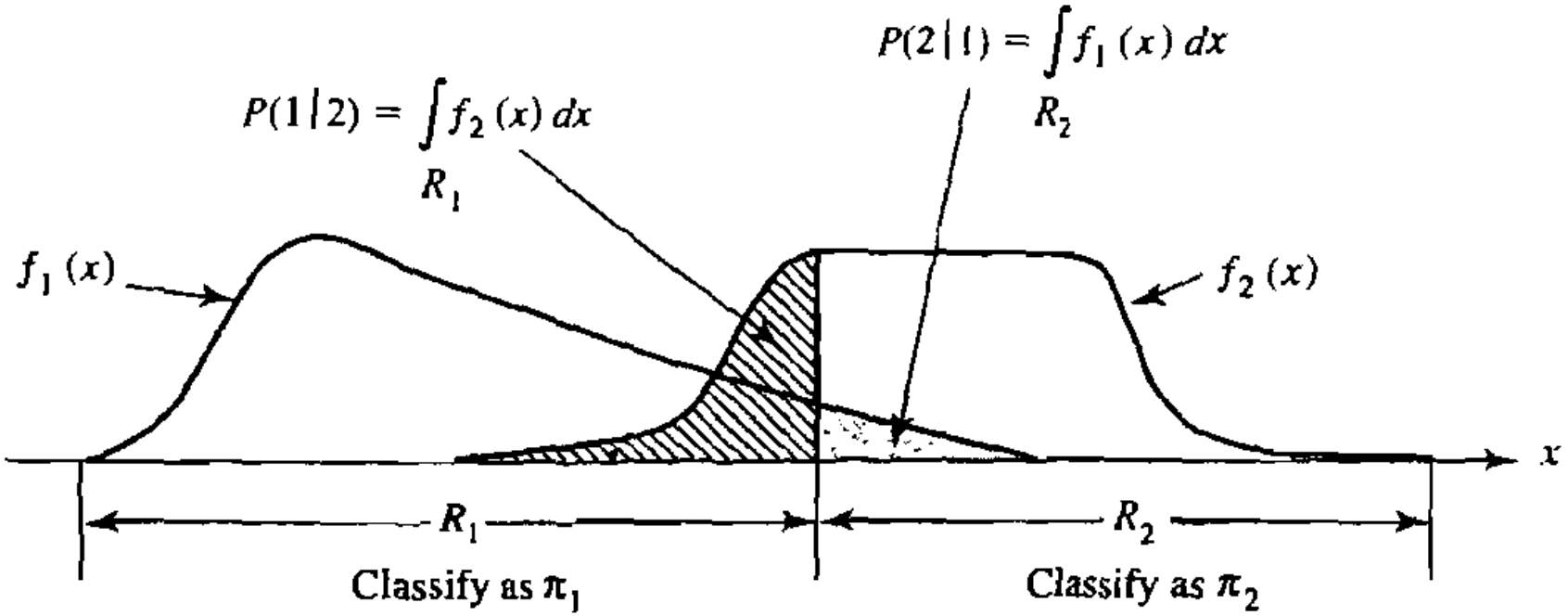


Figure 11.3 Misclassification probabilities for hypothetical classification regions when $p = 1$.

Figure: Figure 11.3 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern). Visualization is for $p = 1$ variable.

Incorporating Prior Probabilities

Let p_1 and p_2 denote the prior probabilities that an object belongs to π_1 and π_2, respectively, with the constraint that $p_1 + p_2 = 1$.

The overall probabilities of the four outcomes have the form

$$
\begin{aligned}
P(\text{correctly classify as } \pi_1) &= P(\boldsymbol{X} \in R_1 | \pi_1) P(\pi_1) = P(1|1) p_1 \\
P(\text{correctly classify as } \pi_2) &= P(\boldsymbol{X} \in R_2 | \pi_2) P(\pi_2) = P(2|2) p_2 \\
P(\text{misclassify } \pi_1 \text{ as } \pi_2) &= P(\boldsymbol{X} \in R_2 | \pi_1) P(\pi_1) = P(2|1) p_1 \\
P(\text{misclassify } \pi_2 \text{ as } \pi_1) &= P(\boldsymbol{X} \in R_1 | \pi_2) P(\pi_2) = P(1|2) p_2
\end{aligned}
$$

Classification Table and Misclassification Costs

In many real world cases, costs of misclassification are not equal:

- π_1 and π_2 are diseased and healthy
- π_1 and π_2 are guilty and not guilty
- π_1 and π_2 are buy and not buy stock

We can make a cost matrix to tabulate our misclassification costs:

		Classify as:	
		π_1	π_2
Truth:	π_1	0	$c(2\|1)$
	π_2	$c(1\|2)$	0

The expected cost of misclassification (ECM) is defined as

$$\text{ECM} = c(2|1)P(2|1)p_1 + c(1|2)P(1|2)p_2$$

Classification Rule (Region) Minimizing ECM

The R_1 and R_2 that minimize the ECM are defined via the inequalities:

$$R_1 : \quad \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$$

$$R_2 : \quad \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)$$

If $c(1|2) = c(2|1)$, then we are classifying via posterior probabilities.

If $c(1|2) = c(2|1)$ and $p_1 = p_2$, then the classification rule reduces to

$$R_1 : \quad \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq 1$$

$$R_2 : \quad \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < 1$$

Two Multivariate Normal Populations

MVN Two Population Classification Problem

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ denote a random vector and let

- $f_1(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ denote the pdf for population π_1
- $f_2(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ denote the pdf for population π_2

Problem: Given a realization $\boldsymbol{X} = \mathbf{x}$, we want to assign $\mathbf{x}$ to π_1 or π_2.

We want to find some classification rule to determine whether a realization $\boldsymbol{X} = \mathbf{x}$ should be assigned to population π_1 or π_2.

Classification Rule Minimizing ECM

The multivariate normal densities have the form

$$f_k(\mathbf{x}) = (2\pi)^{-p/2}|\mathbf{\Sigma}|^{-1/2}\exp\{-(1/2)(\mathbf{x}-\boldsymbol{\mu}_k)'\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

for $k \in \{1,2\}$, which implies that

$$f^* = \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} = \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)'\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)'\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right\}$$

$$= \exp\left\{(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)'\mathbf{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)'\mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1+\boldsymbol{\mu}_2)\right\}$$

The R_1 and R_2 that minimize the ECM are defined via the inequalities:

$$R_1: \quad \log(f^*) \geq \log\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$$

$$R_2: \quad \log(f^*) < \log\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$$

Classification Rule in Practice

The rule on the previous slide depends on the population parameters $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{\Sigma}$, which are often unknown in practice.

Given n_1 independent observations from π_1 and n_2 independent observations from π_2, we can estimate the needed parameters:

$$\hat{\boldsymbol{\mu}}_1 = \bar{\mathbf{x}}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{x}_{i(1)} \qquad \text{and} \qquad \hat{\boldsymbol{\mu}}_2 = \bar{\mathbf{x}}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2}\mathbf{x}_{i(2)}$$

$$\hat{\boldsymbol{\Sigma}} = \mathbf{S}_p = \frac{1}{n_1+n_2-2}\left[\sum_{i=1}^{n_1}(\mathbf{x}_{i(1)}-\bar{\mathbf{x}}_1)(\mathbf{x}_{i(1)}-\bar{\mathbf{x}}_1)' + \sum_{i=1}^{n_2}(\mathbf{x}_{i(2)}-\bar{\mathbf{x}}_2)(\mathbf{x}_{i(2)}-\bar{\mathbf{x}}_2)'\right]$$

The estimated classification rule replaces f^* with its sample estimate:

$$\hat{f}^* = \exp\left\{(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)'\mathbf{S}_p^{-1}\mathbf{x} - \frac{1}{2}(\bar{\mathbf{x}}_1-\bar{\mathbf{x}}_2)'\mathbf{S}_p^{-1}(\bar{\mathbf{x}}_1+\bar{\mathbf{x}}_2)\right\}$$

Classification Rule in Practice (continued)

If $\nu = \left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right) = 1$, then the rule becomes

$$R_1: \quad \hat{y} \geq \hat{m}$$
$$R_2: \quad \hat{y} < \hat{m}$$

where

$$\hat{y} = \hat{\mathbf{a}}'\mathbf{x} \qquad \text{and} \qquad \hat{m} = \frac{1}{2}(\bar{y}_1 + \bar{y}_2)$$

with $\hat{\mathbf{a}}' = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)'\mathbf{S}_p^{-1}$, $\bar{y}_1 = \hat{\mathbf{a}}'\bar{\mathbf{x}}_1$, and $\bar{y}_2 = \hat{\mathbf{a}}'\bar{\mathbf{x}}_2$

Scale of $\hat{\mathbf{a}}$ is not uniquely determined, so normalize $\hat{\mathbf{a}}$ using either:

1. $\hat{\mathbf{a}}^* = \hat{\mathbf{a}}/\|\hat{\mathbf{a}}\|$ (unit length)
2. $\hat{\mathbf{a}}^* = \hat{\mathbf{a}}/\hat{a}_1$ (first element 1)

Fisher's Linear Discriminant Function

R. A. Fisher arrived at the decision rule on the previous slide using an entirely different argument.

Fisher considered finding the linear combination $Y = \mathbf{a}'\boldsymbol{X}$ that best separates the groups:

$$\text{separation} = \frac{|\bar{y}_1 - \bar{y}_2|}{s_y}$$

where

- $\bar{y}_1$ is the mean of the Y scores for the observations from π_1
- $\bar{y}_2$ is the mean of the Y scores for the observations from π_2
- $s_y^2 = \frac{\sum_{i=1}^{n_1}(y_{i(1)}-\bar{y}_1)^2+\sum_{i=1}^{n_2}(y_{i(2)}-\bar{y}_2)^2}{n_1+n_2-2}$ is the pooled variance

Fisher's Linear Discriminant Function (continued)

Setting $\hat{\mathbf{a}}' = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)'\mathbf{S}_p^{-1}$ maximizes the separation

$$\begin{aligned}
\text{separation}^2 &= \frac{(\bar{y}_1 - \bar{y}_2)^2}{s_y^2} \\
&= \frac{(\hat{\mathbf{a}}'\bar{\mathbf{x}}_1 - \hat{\mathbf{a}}'\bar{\mathbf{x}}_2)^2}{\hat{\mathbf{a}}'\mathbf{S}_p\hat{\mathbf{a}}} \\
&= \frac{(\hat{\mathbf{a}}'\mathbf{d})^2}{\hat{\mathbf{a}}'\mathbf{S}_p\hat{\mathbf{a}}} \\
&= \mathbf{d}'\mathbf{S}_p^{-1}\mathbf{d} \\
&= D^2
\end{aligned}$$

overall all possible $\mathbf{a}$ vectors, where $\mathbf{d} = \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2$.

Visualizing Fisher's Linear Discriminant Function

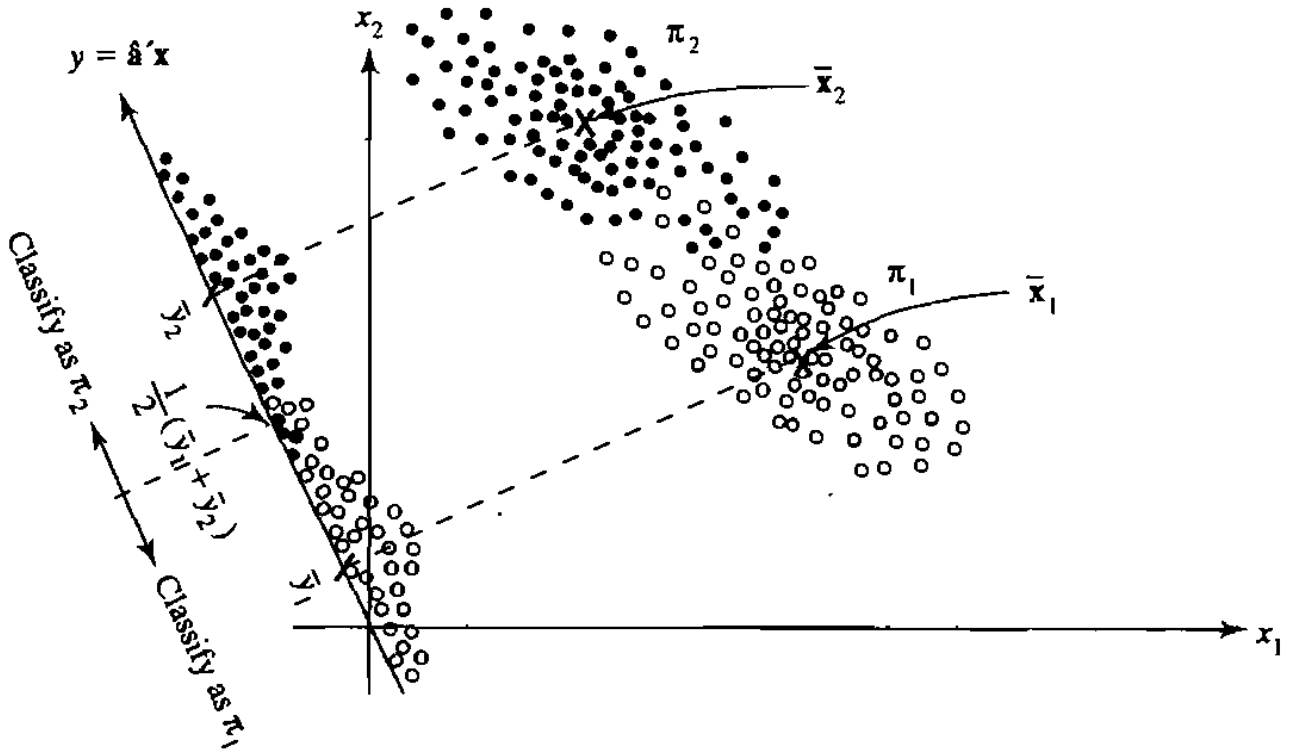


Figure 11.5 A pictorial representation of Fisher's procedure for two populations with $p = 2$.

Figure: Figure 11.5 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern). Visualization is for $p = 2$ variables.

MVN Two Population Classification Problem ($\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$)

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ denote a random vector and let

- $f_1(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ denote the pdf for population π_1
- $f_2(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ denote the pdf for population π_2

Problem: Given a realization $\boldsymbol{X} = \mathbf{x}$, we want to assign $\mathbf{x}$ to π_1 or π_2.

We want to find some classification rule to determine whether a realization $\boldsymbol{X} = \mathbf{x}$ should be assigned to population π_1 or π_2.

Classification Rule Minimizing ECM ($\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$)

The multivariate normal densities have the form

$$f_k(\mathbf{x}) = (2\pi)^{-p/2}|\boldsymbol{\Sigma}_k|^{-1/2}\exp\{-(1/2)(\mathbf{x}-\boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\}$$

for $k \in \{1,2\}$, which implies that

$$f^* = \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} = \left(\frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_2|}\right)^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)'\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right\}$$

The R_1 and R_2 that minimize the ECM are defined via the inequalities:

$$R_1: \quad \log(f^*) \geq \log\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$$

$$R_2: \quad \log(f^*) < \log\left[\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right)\right]$$

Classification Rule in Practice ($\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$)

The rule on the previous slide depends on the population parameters $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\Sigma}_1$, and $\boldsymbol{\Sigma}_2$, which are often unknown in practice.

Given n_1 independent observations from π_1 and n_2 independent observations from π_2, we can estimate the needed parameters:

$$\hat{\boldsymbol{\mu}}_1 = \bar{\mathbf{x}}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\mathbf{x}_{i(1)} \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_1 = \mathbf{S}_1 = \frac{1}{n_1-1}\sum_{i=1}^{n_1}(\mathbf{x}_{i(1)}-\bar{\mathbf{x}}_1)(\mathbf{x}_{i(1)}-\bar{\mathbf{x}}_1)'$$

$$\hat{\boldsymbol{\mu}}_2 = \bar{\mathbf{x}}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2}\mathbf{x}_{i(2)} \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_2 = \mathbf{S}_2 = \frac{1}{n_2-1}\sum_{i=1}^{n_2}(\mathbf{x}_{i(2)}-\bar{\mathbf{x}}_2)(\mathbf{x}_{i(2)}-\bar{\mathbf{x}}_2)'$$

The estimated classification rule replaces f^* with its sample estimate:

$$\hat{f}^* = \left(\frac{|\mathbf{S}_1|}{|\mathbf{S}_2|}\right)^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_1)'\mathbf{S}_1^{-1}(\mathbf{x}-\bar{\mathbf{x}}_1) + \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_2)'\mathbf{S}_2^{-1}(\mathbf{x}-\bar{\mathbf{x}}_2)\right\}$$

Classification Rule in Practice ($\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$), continued

Note that we can write

$$\begin{aligned}\log(\hat{f}^*) &= \log\left[\left(\frac{|\mathbf{S}_1|}{|\mathbf{S}_2|}\right)^{-1/2} e^{-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_1)'\mathbf{S}_1^{-1}(\mathbf{x}-\bar{\mathbf{x}}_1)+\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_2)'\mathbf{S}_2^{-1}(\mathbf{x}-\bar{\mathbf{x}}_2)}\right] \\ &= \hat{y} - \hat{m}\end{aligned}$$

where

$$\hat{y} = -\frac{1}{2}\mathbf{x}'(\mathbf{S}_1^{-1} - \mathbf{S}_2^{-1})\mathbf{x} + (\bar{\mathbf{x}}_1'\mathbf{S}_1^{-1} - \bar{\mathbf{x}}_2'\mathbf{S}_2^{-1})\mathbf{x}$$

$$\hat{m} = \frac{1}{2}\log\left(\frac{|\mathbf{S}_1|}{|\mathbf{S}_2|}\right) + \frac{1}{2}(\bar{\mathbf{x}}_1'\mathbf{S}_1^{-1}\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2'\mathbf{S}_2^{-1}\bar{\mathbf{x}}_2)$$

$\hat{y}$ is a quadratic function of $\mathbf{x}$, so this a quadratic classification rule.

Caution: Quadratic Classification of Non-Normal Data

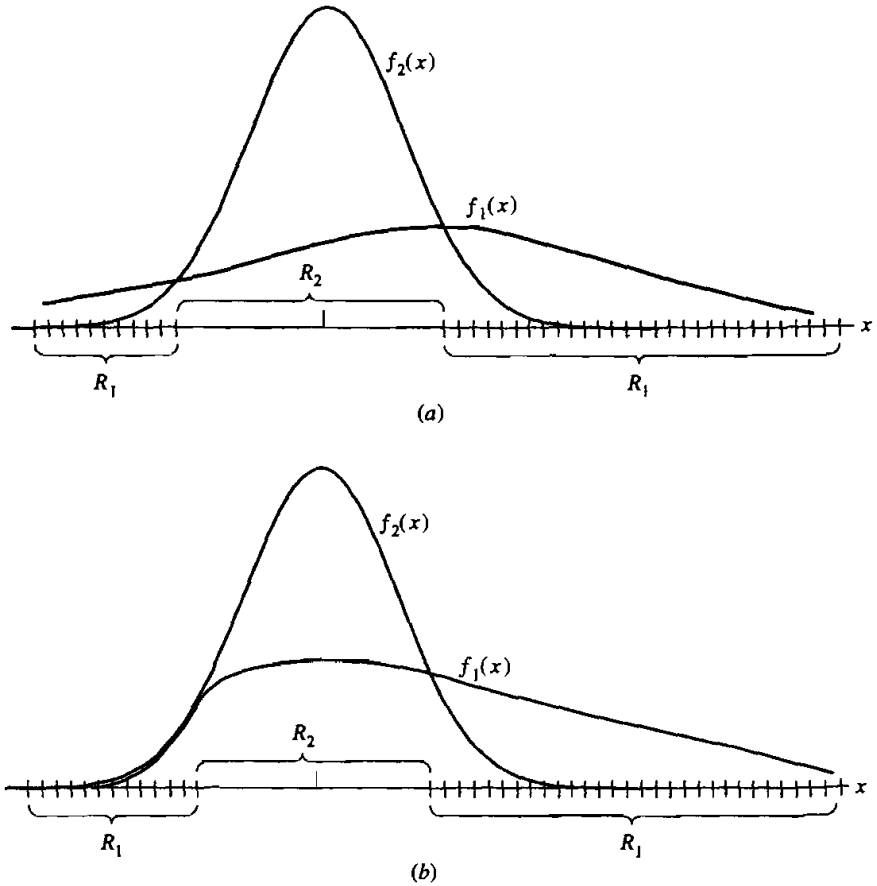


Figure 11.6 Quadratic rules for (a) two normal distribution with unequal variances and (b) two distributions, one of which is nonnormal—rule not appropriate.

Figure: Figure 11.6 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern). Visualization is for $p = 1$ variable.

Evaluating Classification Functions

Quantifying the Quality of a Classification Rule

To determine if a classification rule is “good” we can examine the error rates, i.e., misclassification probabilities.

The population parameters are unknown in practice, so we focus on approaches that can estimate the error rates from the observed data.

We want our classification rule to cross-validate to new data, so we consider cross-validation procedures.

Total Probability of Misclassification

The Total Probability of Misclassification (TPM) is defined as

$$\text{TPM}(R_1, R_2) = p_1 \int_{R_2} f_1(\mathbf{x})d\mathbf{x} + p_2 \int_{R_1} f_2(\mathbf{x})d\mathbf{x}$$

for any classification rule (region) that partitions $\Omega = R_1 \cup R_2$.

The Optimum Error Rate (OER) is the minimum possible value of TPM

$$\text{OER} = \min_{R_1, R_2} \text{TPM}(R_1, R_2) \quad \text{subject to} \quad \Omega = R_1 \cup R_2$$

which is obtained when $R_1 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} \geq \frac{p_2}{p_1}$ and $R_2 : \frac{f_1(\mathbf{x})}{f_2(\mathbf{x})} < \frac{p_2}{p_1}$.

- If $c(1|2) = c(2|1)$, minimizing TPM is same as minimizing ECM

Actual Error Rate

The error rates on the previous slide require knowledge of the (typically unknown) parameters that define the densities $f_1(\cdot)$ and $f_2(\cdot)$.

- Example: For LDA, calculating OER requires $\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, and $\boldsymbol{\Sigma}$

The Actual Error Rate (AER) is defined using the sample estimates

$$\text{AER}(\hat{R}_1, \hat{R}_2) = p_1 \int_{\hat{R}_2} f_1(\mathbf{x})d\mathbf{x} + p_2 \int_{\hat{R}_1} f_2(\mathbf{x})d\mathbf{x}$$

where $\hat{R}_1$ and $\hat{R}_2$ denote estimates from samples sizes n_1 and n_2.

Apparent Error Rate

The Apparent Error Rate (APER) is an—optimistic—estimate of AER.

- Estimates the AER using the observed (training) sample of data

The confusion matrix for a sample of data is

		Classified as: π_1	π_2	
Truth:	π_1	n_{C1}	n_{M1}	n_1
	π_2	n_{M2}	n_{C2}	n_2

where

- n_{Ck} is the number correctly classified in population $k \in \{1, 2\}$
- $n_{M1} = n_1 - n_{C1}$ is the number from π_1 that are misclassified
- $n_{M2} = n_2 - n_{C2}$ is the number from π_2 that are misclassified

Apparent Error Rate (continued)

Given a sample of data with confusion matrix

		Classified as:		
		π_1	π_2	
Truth:	π_1	n_{C1}	n_{M1}	n_1
	π_2	n_{M2}	n_{C2}	n_2

the APER is calculated as

$$\text{APER} = \frac{n_{M1} + n_{M2}}{n_1 + n_2}$$

which is the total proportion of misclassified sample observations.

Leave-One-Out (Ordinary) Cross-Validation

Lachenbruch proposed a better approach to estimate the AER:

1. Population 1 (for $i = 1, \ldots, n_1$)
   (a) Hold out the i-th observation from π_1 and build classification rule
   (b) Use classification rule from Step 1(a) to classify the i-th observation
2. Population 2 (for $i = 1, \ldots, n_2$)
   (a) Hold out the i-th observation from π_2 and build classification rule
   (b) Use classification rule from Step 2(a) to classify the i-th observation

An (almost) unbiased estimate of the expected AER is given by

$$\hat{E}(\text{AER}) = \frac{n_{M1}^* + n_{M2}^*}{n_1 + n_2}$$

where n_{M1}^* and n_{M2}^* are the number of misclassified observations using the above "leave-one-out"' procedure.

Revisiting Linear Discriminant Analysis

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ denote a random vector and let

- $f_1(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$ denote the pdf for population π_1
- $f_2(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$ denote the pdf for population π_2

Reminder: assuming that $\left(\frac{c(1|2)}{c(2|1)}\right)\left(\frac{p_2}{p_1}\right) = 1$, the classification rule is

$$\begin{aligned} R_1 : &\quad Y \geq m \\ R_2 : &\quad Y < m \end{aligned}$$

where

$$Y = \mathbf{a}'\boldsymbol{X} \qquad \text{and} \qquad m = \frac{1}{2}(\mu_{Y_1} + \mu_{Y_2})$$

with $\mathbf{a}' = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}$, $\mu_{Y_1} = \mathbf{a}'\boldsymbol{\mu}_1$, and $\mu_{Y_2} = \mathbf{a}'\boldsymbol{\mu}_2$

Revisiting Linear Discriminant Analysis (continued)

$Y = \mathbf{a}'\boldsymbol{X} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}\boldsymbol{X}$ is a linear function of $\boldsymbol{X}$, so ...

- $\mu_{Y_1} = \mathbf{a}'\boldsymbol{\mu}_1 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1$
- $\mu_{Y_2} = \mathbf{a}'\boldsymbol{\mu}_2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2$
- $\sigma^2_Y = \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = \Delta^2$

And since $\boldsymbol{X}$ is multivariate normal, we have that

$$Y \sim \begin{cases} \mathrm{N}(\mu_{Y_1}, \Delta^2) & \text{if from } \pi_1 \\ \mathrm{N}(\mu_{Y_2}, \Delta^2) & \text{if from } \pi_2 \end{cases}$$

i.e., Y is univariate normal with population dependent mean.

Visualizing Misclassification in LDA

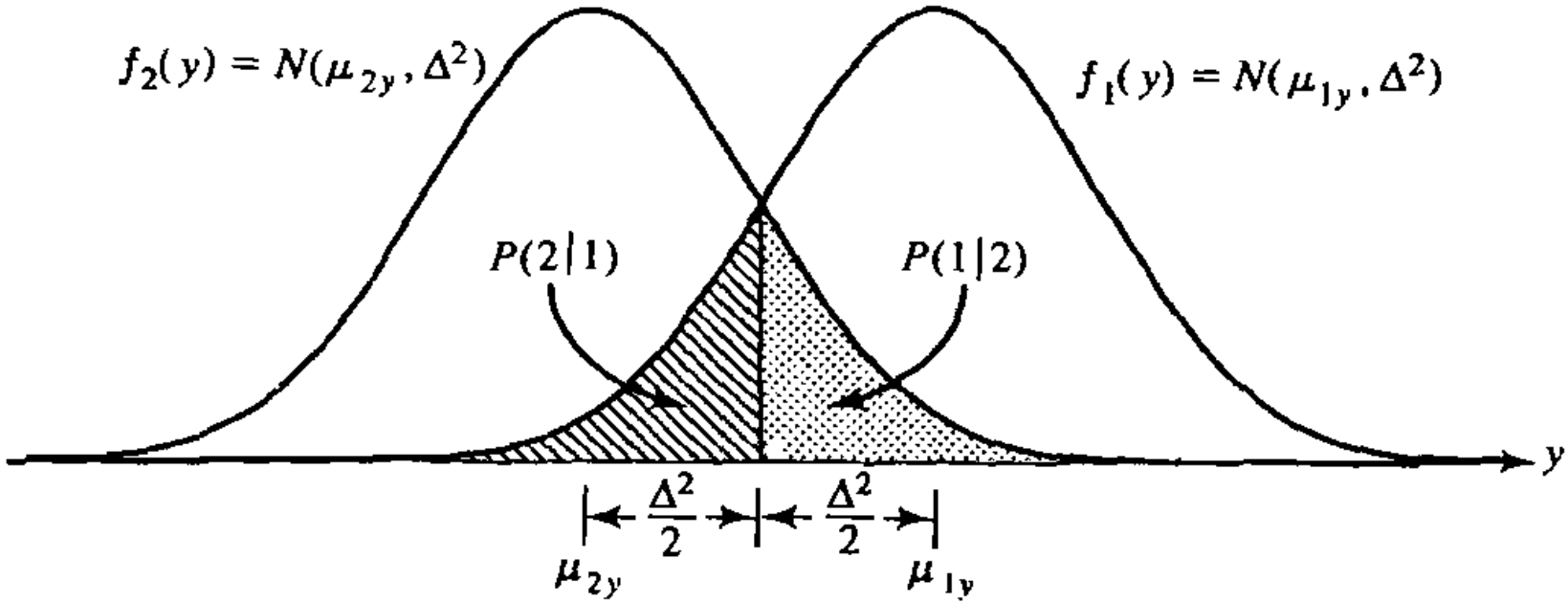


Figure 11.7 The misclassification probabilities based on Y.

Figure: Figure 11.7 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern).

Calculating Misclassification in LDA (classify π_1 as π_2)

Defining $m = (1/2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$, we have that

$$\begin{aligned} P(\text{misclassify } \pi_1 \text{ as } \pi_2) &= P(\boldsymbol{X} \in R_2|\pi_1) = P(2|1) \\ &= P\left(Y < m\right) \\ &= P\left(\frac{Y - \mu_{Y_1}}{\sigma_Y} < \frac{m - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1}{\Delta}\right) \\ &= P\left(Z < \frac{-(1/2)\Delta^2}{\Delta}\right) \\ &= \boldsymbol{\Phi}(-\Delta/2) \end{aligned}$$

where $\boldsymbol{\Phi}(\cdot)$ denotes the CDF of the standard normal distribution.

Calculating Misclassification in LDA (classify π_2 as π_1)

Defining $m = (1/2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$, we have that

$$\begin{aligned} P(\text{misclassify } \pi_2 \text{ as } \pi_1) &= P(\boldsymbol{X} \in R_1 | \pi_2) = P(1|2) \\ &= P\left(Y \geq m\right) \\ &= P\left(\frac{Y - \mu_{Y_2}}{\sigma_Y} \geq \frac{m - (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2}{\Delta}\right) \\ &= P\left(Z \geq \frac{(1/2)\Delta^2}{\Delta}\right) \\ &= 1 - \boldsymbol{\Phi}(\Delta/2) = \boldsymbol{\Phi}(-\Delta/2) \end{aligned}$$

where $\boldsymbol{\Phi}(\cdot)$ denotes the CDF of the standard normal distribution.

Optimum Error Rate for Linear Discriminant Analysis

For the LDA classification rule, we have that

$$\begin{aligned}
\text{OER} &= \min_{R_1,R_2} \text{TPM}(R_1, R_2) \\
&= \frac{1}{2}P(\text{misclassify } \pi_1 \text{ as } \pi_2) + \frac{1}{2}P(\text{misclassify } \pi_2 \text{ as } \pi_1) \\
&= \frac{1}{2}\boldsymbol{\Phi}(-\Delta/2) + \frac{1}{2}[1 - \boldsymbol{\Phi}(\Delta/2)] \\
&= \boldsymbol{\Phi}(-\Delta/2)
\end{aligned}$$

so the OER is a function of the Δ effect size

$$\Delta = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}$$

which is distance measure between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$.

Classifying $g \geq 2$ Populations

The g Population Classification Problem

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ denote a random vector and let $f_k(\mathbf{x})$ denote the probability density function (pdf) for population π_k for $k \in \{1, \ldots, g\}$.

Problem: Given a realization $\boldsymbol{X} = \mathbf{x}$, we want to assign $\mathbf{x}$ to a π_k.

We want to find some classification rule to determine whether a realization $\boldsymbol{X} = \mathbf{x}$ should be assigned to population π_1, π_2, ..., or π_g.

Classification Rule with $g \geq 2$ Populations

Let Ω denote the sample space, i.e., all possible values of $\mathbf{x}$, and

- $R_1 \subset \Omega$ is the subset of Ω for which we classify $\mathbf{x}$ as π_1
- $R_2 \subset \Omega$ is the subset of Ω for which we classify $\mathbf{x}$ as π_2

  $\vdots$

- $R_g \subset \Omega$ is the subset of Ω for which we classify $\mathbf{x}$ as π_g

$\Omega = R_1 \cup R_2 \cup \cdots \cup R_g$ and $R_k \cap R_\ell = \emptyset$ for all $k \neq \ell$.

- The classification rule partitions the sample space
- The classification regions are mutually exclusive

Visualizing a Classification Rule: $g = 3$ Populations

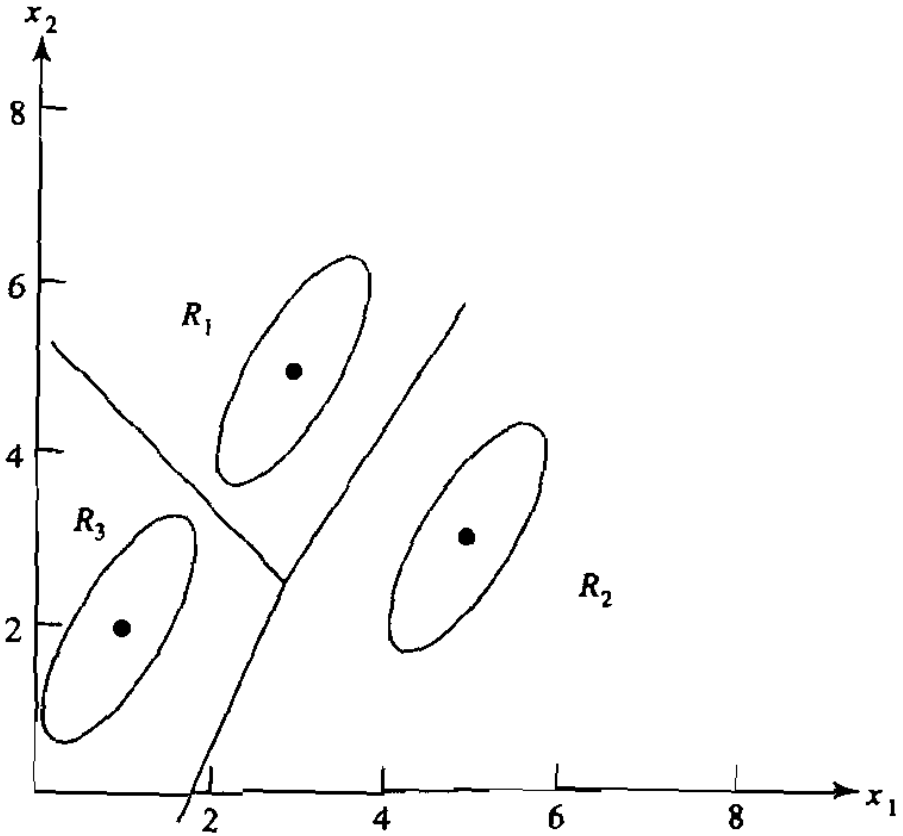


Figure 11.10 The classification regions R_1, R_2, and R_3 for the linear minimum TPM rule $(p_1 = \frac{1}{4}, p_2 = \frac{1}{2}, p_3 = \frac{1}{4})$.

Figure: Figure 11.10 from Applied Multivariate Statistical Analysis, 6th Ed (Johnson & Wichern). Visualization is for $p = 2$ variables.

Probability and Cost of Misclassification

The conditional probability $P(\ell|k)$ of classifying an object as π_ℓ when the object really belongs to π_k is given by

$$P(\ell|k) = P(\boldsymbol{X} \in R_\ell|\pi_k) = \int_{R_\ell} f_k(\mathbf{x})\mathrm{d}\mathbf{x}$$

for all $k \neq \ell$ with $k, \ell \in \{1, \ldots, g\}$.

Note that $P(k|k) = 1 - \sum_{\ell \neq k} P(\ell|k)$ by definition.

Let $c(\ell|k)$ denote the cost of allocating an object to π_ℓ when the object really belongs to π_k, and let p_k denote the prior probability of π_k.

Expected Cost of Misclassification (revisited)

The conditional expected cost of misclassifying an object from π_k is

$$\text{ECM}(k) = \sum_{\ell \neq k} P(\ell|k)c(\ell|k)$$

Incorporating the prior probabilities, the overall ECM is given by

$$\text{ECM} = \sum_{k=1}^{g} p_k \text{ECM}(k) = \sum_{k=1}^{g} p_k \left[\sum_{\ell \neq k} P(\ell|k)c(\ell|k) \right]$$

Minimum ECM Classification Rule

The classification regions $\{R_1, R_2, \ldots, R_g\}$ that minimize the ECM are defined by allocating $\boldsymbol{X} = \mathbf{x}$ to the population π_k that minimizes

$$\sum_{\ell \neq k} p_\ell f_\ell(\mathbf{x}) c(k|\ell)$$

To understand the logic of the classification rule, suppose that we have equal costs, i.e., $c(\ell|k) = c(k|\ell) = 1$ for all $k, \ell \in \{1, \ldots, g\}$

- We allocate $\mathbf{x}$ to the population π_k that minimizes $\sum_{\ell \neq k} p_\ell f_\ell(\mathbf{x})$
- Minimizing $\sum_{\ell \neq k} p_\ell f_\ell(\mathbf{x})$ is the same as maximizing $p_k f_k(\mathbf{x})$
- Allocate $\mathbf{x}$ to population π_k if $p_k f_k(\mathbf{x}) > p_\ell f_\ell(\mathbf{x})$ for all $\ell \neq k$
- This is equivalent to maximizing the posterior probability $P(\pi_k|\mathbf{x})$

Overview of Fisher's Approach

Fisher developed his discriminant analysis for $g > 2$ populations.

Idea: find a small number of linear combinations (e.g., $\mathbf{a}_1'\mathbf{x}$, $\mathbf{a}_2'\mathbf{x}$, $\mathbf{a}_3'\mathbf{x}$) that best separate the groups.

Offers a simple and useful procedure for classification, which also provides nice visualizations.

- Plot the linear combinations to visualize the discriminants

Assumptions of Fisher's Discriminant Analysis

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ denote a random vector and let $f_k(\mathbf{x}) \sim (\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$ denote the pdf for population π_k.

- Note the homogeneity of covariance matrix assumption
- Do not need the multivariate normality assumption

Let $\bar{\boldsymbol{\mu}} = \frac{1}{g}\sum_{k=1}^{g} \boldsymbol{\mu}_k$ denote the mean of the combined populations, and

$$\mathbf{B}_\mu = \sum_{k=1}^{g} (\boldsymbol{\mu}_k - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_k - \bar{\boldsymbol{\mu}})'$$

denote "Between" sum-of-squares and crossproducts (SSCP) matrix.

Properties of a Linear Combination

Define new variable $Y = \mathbf{a}'\boldsymbol{X}$ which has properties

$$E(Y|\pi_k) = \mathbf{a}'E(\boldsymbol{X}|\pi_k) = \mathbf{a}'\boldsymbol{\mu}_k$$
$$V(Y|\pi_k) = \mathbf{a}'V(\boldsymbol{X}|\pi_k)\mathbf{a} = \mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}$$

and note that the overall mean of Y has the form

$$\bar{\mu}_Y = \frac{1}{g}\sum_{k=1}^{g}\mu_{Y_k} = \frac{1}{g}\sum_{k=1}^{g}\mathbf{a}'\boldsymbol{\mu}_k = \mathbf{a}'\bar{\boldsymbol{\mu}}$$

Between versus Within Group Variability

Form the ratio of the between group separation over the variance of Y:

$$
\begin{aligned}
F^* &= \frac{\sum_{k=1}^{g}(\mu_{Y_k} - \bar{\mu}_Y)^2}{\sigma_Y^2} \\
&= \frac{\sum_{k=1}^{g}(\mathbf{a}'\boldsymbol{\mu}_k - \mathbf{a}'\bar{\boldsymbol{\mu}})^2}{\mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}} \\
&= \frac{\mathbf{a}'\left[\sum_{k=1}^{g}(\boldsymbol{\mu}_k - \bar{\boldsymbol{\mu}})(\boldsymbol{\mu}_k - \bar{\boldsymbol{\mu}})'\right]\mathbf{a}}{\mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}} \\
&= \frac{\mathbf{a}'\mathbf{B}_{\mu}\mathbf{a}}{\mathbf{a}'\boldsymbol{\Sigma}\mathbf{a}}
\end{aligned}
$$

Note that higher F^* values relate to more separation between groups.

Population Discriminants

The population k-th discriminant is the linear combination

$$Y_k = \mathbf{a}_k' \boldsymbol{X}$$

where $\mathbf{a}_k$ is proportional to the k-th eigenvector of $\boldsymbol{\Sigma}^{-1}\mathbf{B}_\mu$.

- $k = 1, \ldots, s$ where $s = \min(g-1, p)$

The $\mathbf{a}_k$ are scaled to make the Y_k have unit variance, i.e., $\mathbf{a}_k'\boldsymbol{\Sigma}\mathbf{a}_k = 1$.

- $\mathbf{a}_k'\boldsymbol{\Sigma}\mathbf{a}_\ell = 0$ for $k \neq \ell$

Note that this is only useful if we somehow know the true population parameters $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}$.

Sample Discriminants

The sample estimated "Between" and "Within" SSCP matrices are

$$\mathbf{B} = \sum_{k=1}^{g} (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})' \qquad \text{and} \qquad \mathbf{W} = \sum_{k=1}^{g} \sum_{i=1}^{n_k} (\mathbf{x}_{i(k)} - \bar{\mathbf{x}}_k)(\mathbf{x}_{i(k)} - \bar{\mathbf{x}}_k)'$$

where $\bar{\mathbf{x}}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbf{x}_{i(k)}$ and $\bar{\mathbf{x}} = \frac{1}{g} \sum_{k=1}^{g} \bar{\mathbf{x}}_k$.

The sample k-th discriminant is the linear combination

$$\hat{Y}_k = \hat{\mathbf{a}}_k' \boldsymbol{X}$$

where $\hat{\mathbf{a}}_k$ is proportional to the k-th eigenvector of $\mathbf{W}^{-1}\mathbf{B}$.

The $\hat{\mathbf{a}}_k$ are scaled to make the $\hat{Y}_k$ have unit variance, i.e., $\hat{\mathbf{a}}_k' \hat{\boldsymbol{\Sigma}} \hat{\mathbf{a}}_k = 1$, where $\hat{\boldsymbol{\Sigma}} = \mathbf{S}_p = \frac{1}{n-g}\mathbf{W}$ with $n = \sum_{k=1}^{g} n_k$.

Properties of Population Discriminants

Let $\boldsymbol{Y} = \mathbf{A}'\boldsymbol{X}$ where $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_s]$.

- $\boldsymbol{Y} = (Y_1, \ldots, Y_s)'$ contains the s discriminants
- Columns of $\mathbf{A}$ contain the linear combination weights

The mean of $\boldsymbol{Y}$ is given by

$$E(\boldsymbol{Y}|\pi_k) = \mathbf{A}'E(\boldsymbol{X}|\pi_k) = \mathbf{A}'\boldsymbol{\mu}_k = \boldsymbol{\mu}_{kY}$$

and the covariance matrix for $\boldsymbol{Y}$ is

$$\text{Cov}(\boldsymbol{Y}) = \mathbf{A}'\text{Cov}(\boldsymbol{X}|\pi_k)\mathbf{A} = \mathbf{A}'\boldsymbol{\Sigma}\mathbf{A} = \mathbf{I}_s$$

because the discriminants have unit variance and are uncorrelated.

- Remember: $\mathbf{a}_k'\boldsymbol{\Sigma}\mathbf{a}_\ell = \delta_{k\ell}$ where $\delta_{k\ell}$ is Kronecker's δ

Classifying New Objects with Discriminants

Given a realization $\boldsymbol{X} = \mathbf{x}$, define $\mathbf{y} = \mathbf{A}'\mathbf{x}$ and calculate the distance between the observed $\mathbf{y} = (y_1, \ldots, y_s)'$ and the k-th population mean:

$$D_k = (\mathbf{y} - \boldsymbol{\mu}_{kY})'(\mathbf{y} - \boldsymbol{\mu}_{kY}) = \sum_{\ell=1}^{s}(y_\ell - \mu_{kY_\ell})^2 = \sum_{\ell=1}^{s}[\mathbf{a}_\ell'(\mathbf{x} - \boldsymbol{\mu}_k)]^2$$

where $\boldsymbol{\mu}_{kY} = \mathbf{A}'\boldsymbol{\mu}_k$ and $y_\ell = \mathbf{a}_\ell'\mathbf{x}$ and $\mu_{kY_\ell} = \mathbf{a}_\ell'\boldsymbol{\mu}_k$.

To build a distance using $r \leq s$ discriminants, use

$$D_k^{(r)} = \sum_{\ell=1}^{r}(y_\ell - \mu_{kY_\ell})^2 = \sum_{\ell=1}^{r}[\mathbf{a}_\ell'(\mathbf{x} - \boldsymbol{\mu}_k)]^2$$

and classify $\mathbf{x}$ to the population π_k that minimizes the distance $D_k^{(r)}$.

Classifying New Objects with Sample Discriminants

Given a realization $\boldsymbol{X} = \mathbf{x}$, define $\hat{\mathbf{y}} = \hat{\mathbf{A}}'\mathbf{x}$ and calculate the distance between the observed $\hat{\mathbf{y}} = (\hat{y}_1, \ldots, \hat{y}_s)'$ and the k-th sample mean:

$$\hat{D}_k = (\hat{\mathbf{y}} - \hat{\boldsymbol{\mu}}_{kY})'(\hat{\mathbf{y}} - \hat{\boldsymbol{\mu}}_{kY}) = \sum_{\ell=1}^{s}(\hat{y}_\ell - \hat{\mu}_{kY_\ell})^2 = \sum_{\ell=1}^{s}[\hat{\mathbf{a}}_\ell'(\mathbf{x} - \bar{\mathbf{x}}_k)]^2$$

where $\hat{\boldsymbol{\mu}}_{kY} = \hat{\mathbf{A}}'\bar{\mathbf{x}}_k$ and $\hat{y}_\ell = \hat{\mathbf{a}}_\ell'\mathbf{x}$ and $\hat{\mu}_{kY_\ell} = \hat{\mathbf{a}}_\ell'\bar{\mathbf{x}}_k$.

To build a distance using $r \leq s$ discriminants, use

$$\hat{D}_k^{(r)} = \sum_{\ell=1}^{r}(\hat{y}_\ell - \hat{\mu}_{kY_\ell})^2 = \sum_{\ell=1}^{r}[\hat{\mathbf{a}}_\ell'(\mathbf{x} - \bar{\mathbf{x}}_k)]^2$$

and classify $\mathbf{x}$ to the population π_k that minimizes the distance $\hat{D}_k^{(r)}$.

Relation to MVN Classification Problem

Let $\boldsymbol{X} = (X_1, \ldots, X_p)'$ be a random vector and let $f_k(\mathbf{x}) \sim \mathrm{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denote the pdf for population π_k.

Assuming equal misclassification costs, we allocate $\boldsymbol{X} = \mathbf{x}$ to the population π_k that minimizes $\sum_{\ell \neq k} p_\ell f_\ell(\mathbf{x}) \iff$ maximizes $p_k f_k(\mathbf{x})$.

Equivalent to allocating $\boldsymbol{X} = \mathbf{x}$ to the population π_k that maximizes

$$\begin{aligned} d_k^Q(\mathbf{x}) &= \text{Quadratic discriminant score} \\ &= -\frac{1}{2}\ln(|\boldsymbol{\Sigma}_k|) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \ln(p_k) \\ d_k^L(\mathbf{x}) &= \text{Linear discriminant score} \\ &= \boldsymbol{\mu}_k'\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_k'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln(p_k) \end{aligned}$$

where d_k^L is used when $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$ for all $k \in \{1, \ldots, g\}$.

Relation to MVN Classification Problem (continued)

If we assume that $p_k = 1/g$ for all $k \in \{1, \ldots, g\}$, then

$$d_k^L(\mathbf{x}) = \boldsymbol{\mu}_k' \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k$$

Define the linear combination $y_j = \mathbf{a}_j' \mathbf{x}$, where $\mathbf{a}_j = \boldsymbol{\Sigma}^{-1/2} \mathbf{v}_j$ with $\mathbf{v}_j$ denoting the j-th eigenvector of $\tilde{\mathbf{B}}_\mu = \boldsymbol{\Sigma}^{-1/2} \mathbf{B}_\mu \boldsymbol{\Sigma}^{-1/2}$. Then

$$\begin{aligned} D_k &= \sum_{j=1}^{p} (y_j - \mu_{kY_j})^2 = \sum_{j=1}^{p} [\mathbf{a}_j'(\mathbf{x} - \boldsymbol{\mu}_k)]^2 = (\mathbf{x} - \boldsymbol{\mu}_k)' \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \\ &= -2 d_k^L(\mathbf{x}) + \alpha \end{aligned}$$

where $\alpha = \mathbf{x}' \boldsymbol{\Sigma}^{-1} \mathbf{x}$ is constant across populations.

If $\text{rank}(\tilde{\mathbf{B}}_\mu) = r$, allocating to the population π_k that maximizes $d_k^L(\mathbf{x})$ is equivalent to allocating to the population π_k that minimizes $D_k^{(r)}$.

Fisher's Iris Data Example

Fisher's (or Anderson's) Famous Iris Data

R. A. Fisher published the LDA approach in 1936 and used Edgar Anderson's iris flower dataset as an example.

The dataset consists of measurements of $p = 4$ variables taken from $n_k = 50$ flowers randomly sampled from each of $g = 3$ species.

- Variables: Sepal Length, Sepal Width, Petal Length, Petal Width
- Species: setosa, versicolor, virginica

The goal was/is to build a linear discriminant function that best classifies a new flower into one of the three species.

Fisher's Famous Iris Data in R

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> colMeans(iris[iris$Species=="setosa",1:4])
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.006        3.428        1.462        0.246
> colMeans(iris[iris$Species=="versicolor",1:4])
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       5.936        2.770        4.260        1.326
> colMeans(iris[iris$Species=="virginica",1:4])
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
       6.588        2.974        5.552        2.026
> p <- 4L
> g <- 3L
```

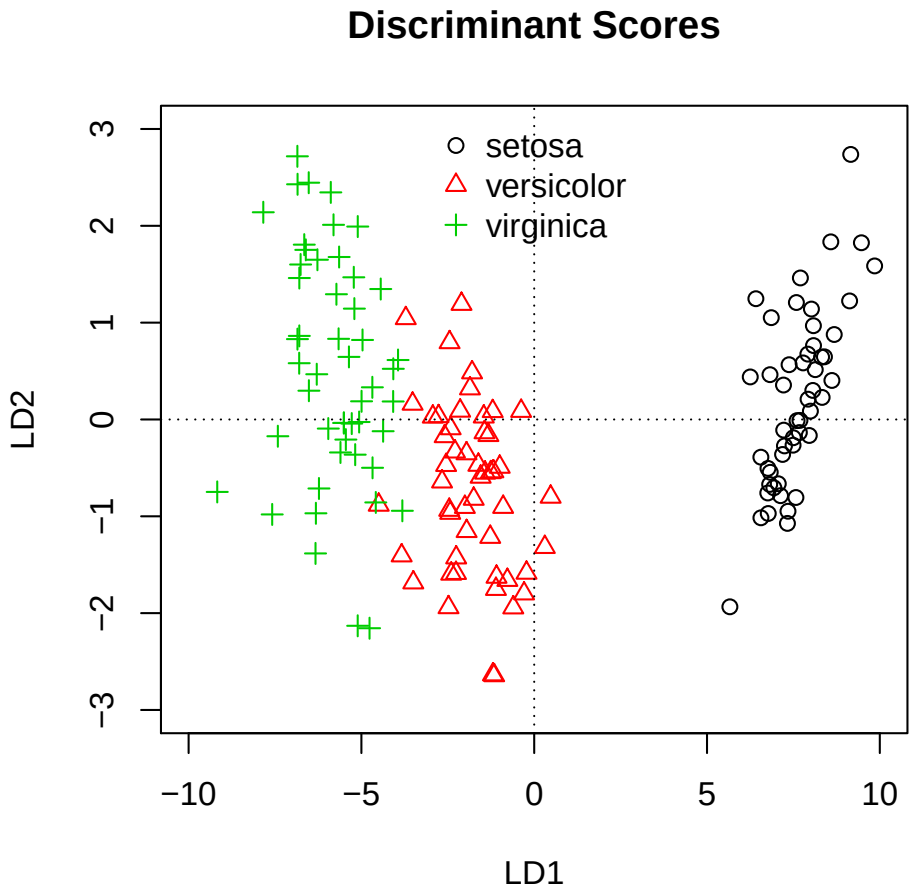
Make Pooled Covariance Matrix

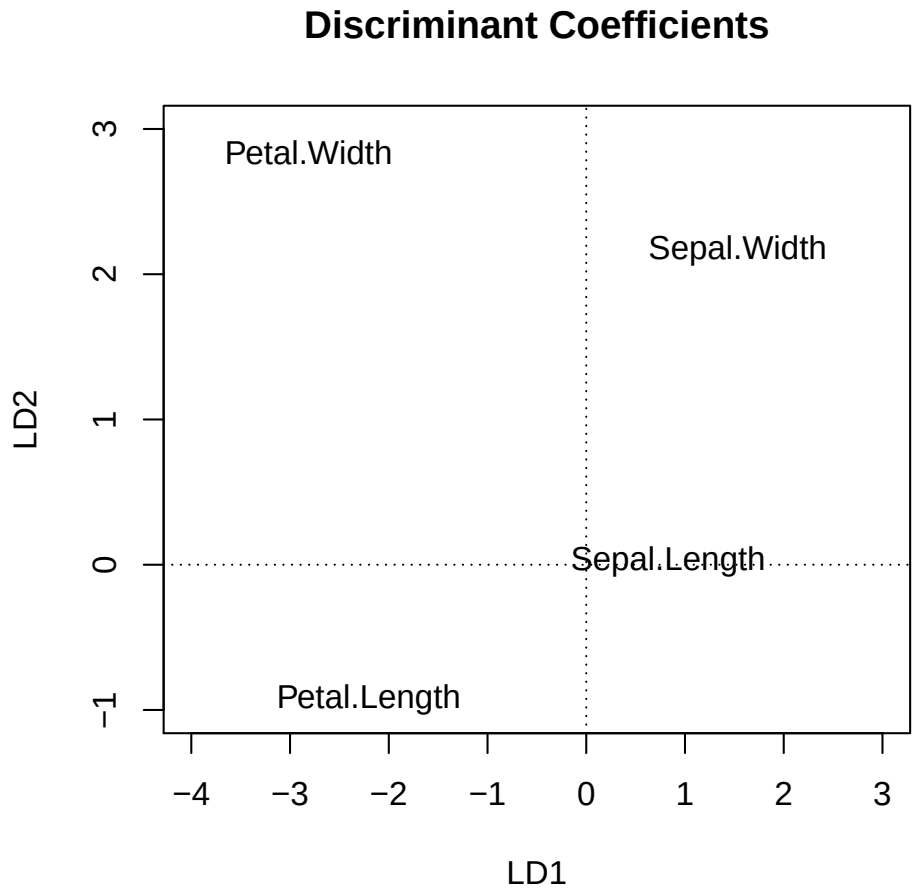
```
# make pooled covariances matrix
> Sp <- matrix(0, p, p)
> nx <- rep(0, g)
> lev <- levels(iris$Species)
> for(k in 1:g){
+     x <- iris[iris$Species==lev[k],1:p]
+     nx[k] <- nrow(x)
+     Sp <- Sp + cov(x) * (nx[k] - 1)
+ }
> Sp <- Sp / (sum(nx) - g)
> round(Sp, 3)
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length        0.265       0.093        0.168       0.038
Sepal.Width         0.093       0.115        0.055       0.033
Petal.Length        0.168       0.055        0.185       0.043
Petal.Width         0.038       0.033        0.043       0.042
```

LDA in R via the lda Function (MASS Package)

```
# fit lda model
> library(MASS)
> ldamod <- lda(Species ~ ., data=iris, prior=rep(1/3, 3))

# check the LDA coefficients/scalings
> ldamod$scaling
                    LD1         LD2
Sepal.Length  0.8293776  0.02410215
Sepal.Width   1.5344731  2.16452123
Petal.Length -2.2012117 -0.93192121
Petal.Width  -2.8104603  2.83918785
> crossprod(ldamod$scaling, Sp) %*% ldamod$scaling
              LD1           LD2
LD1  1.00000e+00 -7.21645e-16
LD2 -7.21645e-16  1.00000e+00

# create the (centered) discriminant scores
> mu.k <- ldamod$means
> mu <- colMeans(mu.k)
> dscores <- scale(iris[,1:p], center=mu, scale=F) %*% ldamod$scaling
> sum((dscores - predict(ldamod)$x)^2)
[1] 1.658958e-28
```

Plot LDA Results: Score and Coefficients

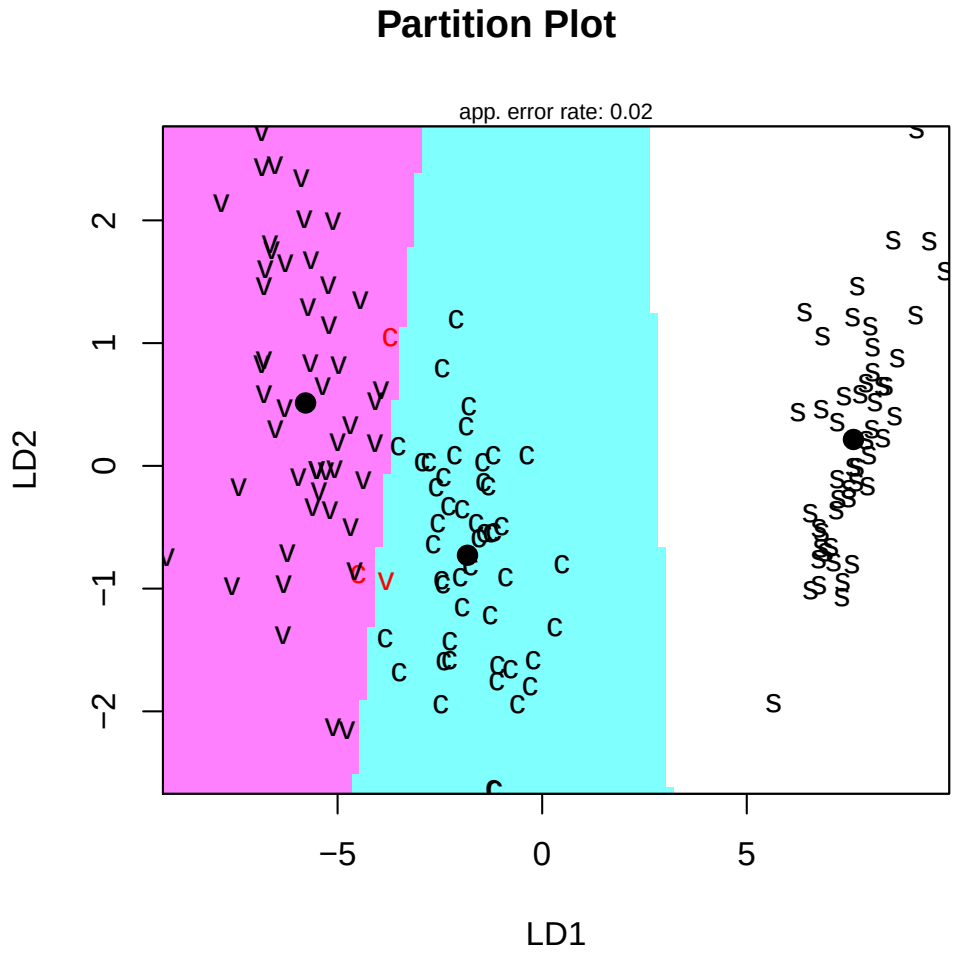




R code for left plot:

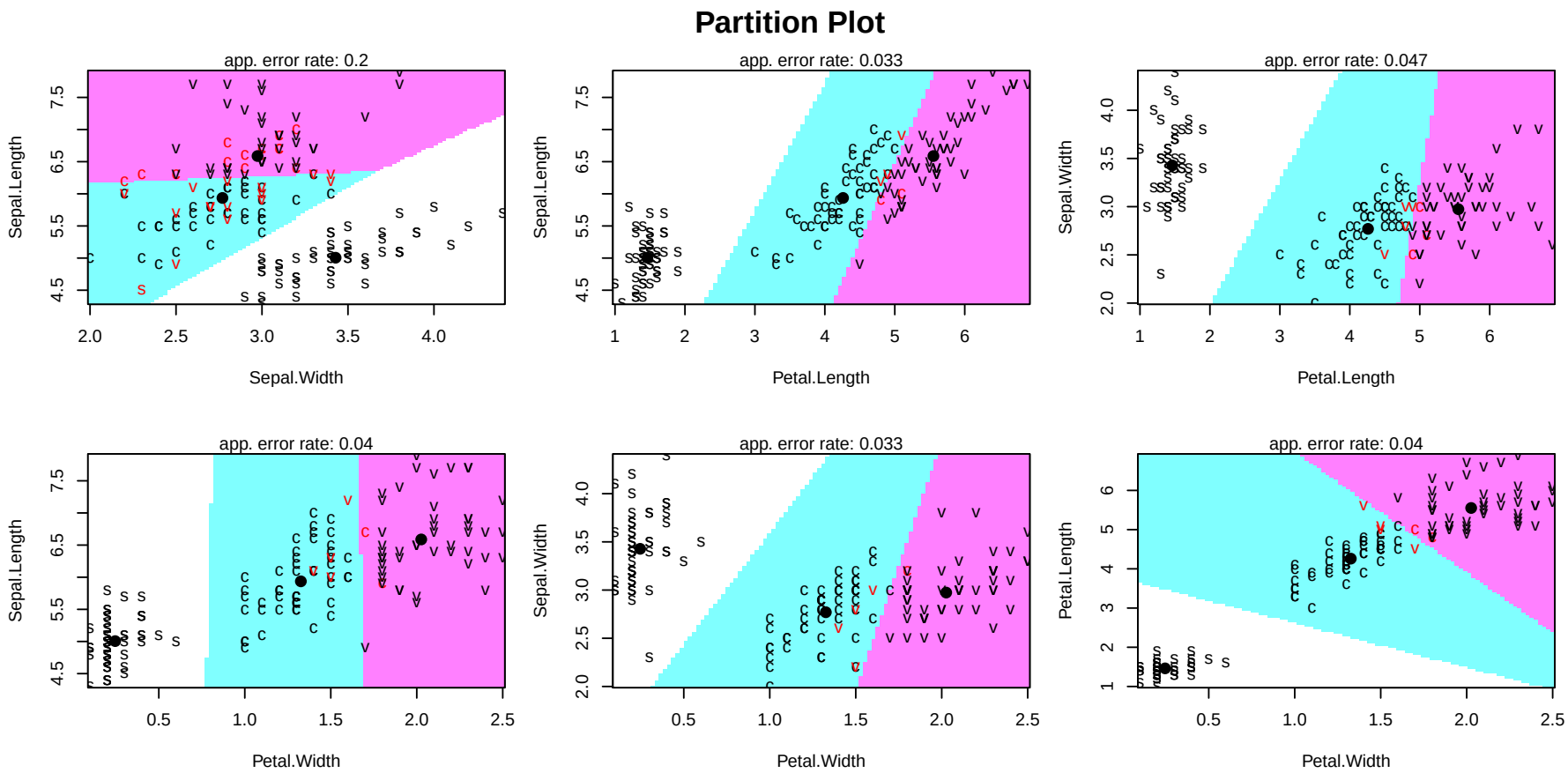
```
plot(dscores, xlab="LD1", ylab="LD2", pch=spid, col=spid,
     main="Discriminant Scores", xlim=c(-10, 10), ylim=c(-3, 3))
legend("top",lev,pch=1:3,col=1:3,bty="n")
```

Plot LDA Results: Discriminant Partitions




```
library(klaR)
species <- factor(rep(c("s","c","v"), each=50))
partimat(x=dscores[,2:1], grouping=species, method="lda")
```

Plot LDA Results: All Pairwise Partitions




```
library(klaR)
species <- factor(rep(c("s","c","v"), each=50))
partimat(x=iris[,1:4], grouping=species, method="lda")
```

APER and Expected AER

```
# make confusion matrix (and APER)
> confusion <- table(iris$Species, predict(ldamod)$class)
> confusion

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49
> n <- sum(confusion)
> aper <- (n - sum(diag(confusion))) / n
> aper
[1] 0.02

# use CV to get expected AER
> ldamodCV <- lda(Species ~ ., data=iris, prior=rep(1/3, 3), CV=TRUE)
> confusionCV <- table(iris$Species, ldamodCV$class)
> confusionCV

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49
> eaer <- (n - sum(diag(confusionCV))) / n
> eaer
[1] 0.02
```

Split Data into Training (70%) and Testing (30%) Sets

```
> # split into separate matrices for each flower
> Xs <- subset(iris, Species=="setosa")
> Xc <- subset(iris, Species=="versicolor")
> Xv <- subset(iris, Species=="virginica")

# split into training and testing
> set.seed(1)
> sid <- sample.int(n=50, size=35)
> cid <- sample.int(n=50, size=35)
> vid <- sample.int(n=50, size=35)
> Xtrain <- rbind(Xs[sid,], Xc[cid,], Xv[vid,])
> Xtest <- rbind(Xs[-sid,], Xc[-cid,], Xv[-vid,])

# fit lda to training and evaluate on testing
> ldatrain <- lda(Species ~ ., data=Xtrain, prior=rep(1/3, 3))
> confusionTest <- table(Xtest$Species, predict(ldatrain, newdata=Xtest)$class)
> confusionTest

            setosa versicolor virginica
  setosa        15          0         0
  versicolor     0         15         0
  virginica      0          1        14
> n <- sum(confusionTest)
> aer <- (n - sum(diag(confusionTest))) / n
> aer
[1] 0.02222222
```

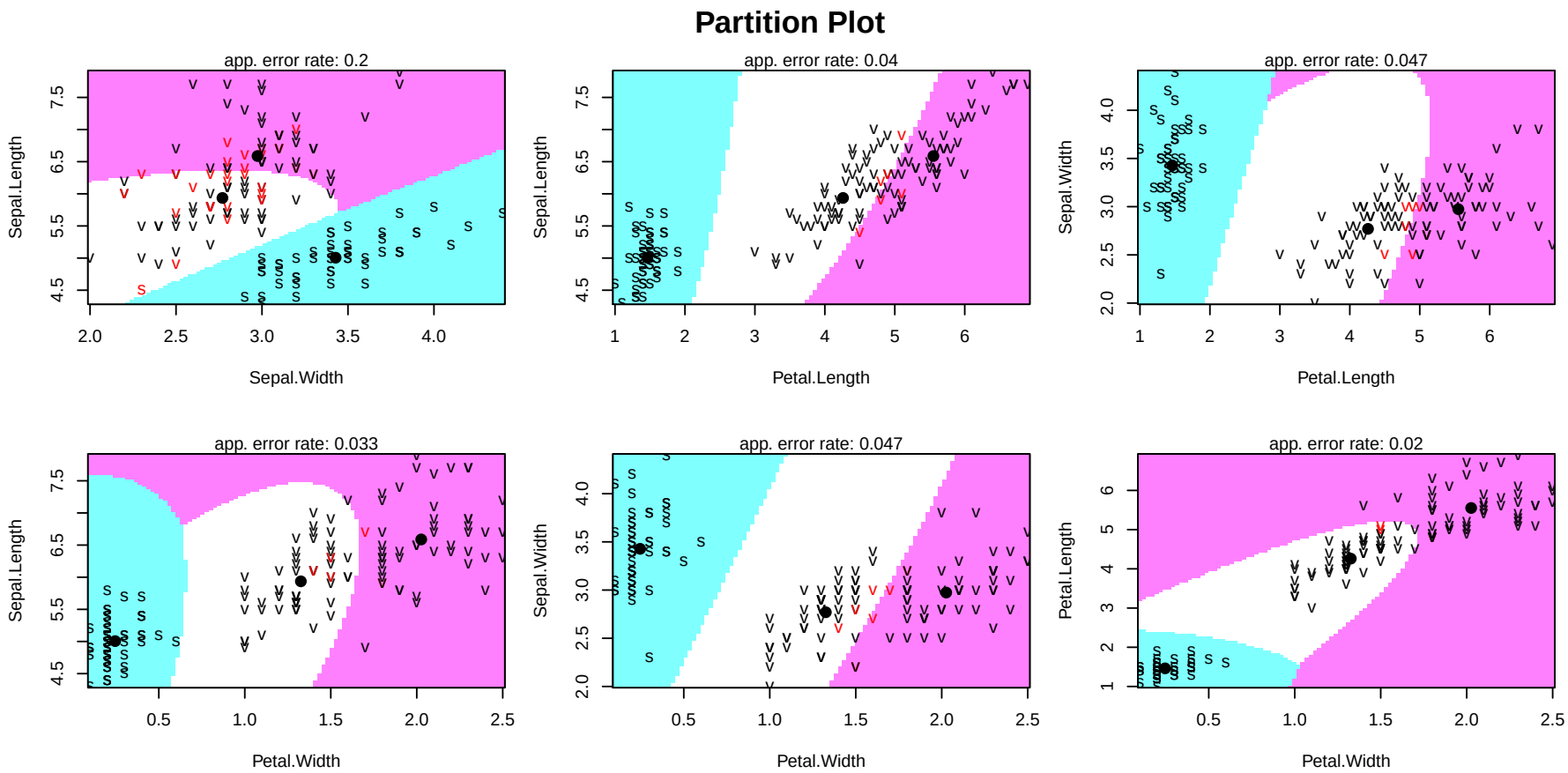
Two-Fold CV with 100 Random 70/30 Splits

```
> nrep <- 100
> aer <- rep(0, nrep)
> set.seed(1)
> for(k in 1:nrep){
+     sid <- sample.int(n=50, size=35)
+     cid <- sample.int(n=50, size=35)
+     vid <- sample.int(n=50, size=35)
+     Xtrain <- rbind(Xs[sid,], Xc[cid,], Xv[vid,])
+     Xtest <- rbind(Xs[-sid,], Xc[-cid,], Xv[-vid,])
+     ldatrain <- lda(Species ~ ., data=Xtrain, prior=rep(1/3, 3))
+     confusionTest <- table(Xtest$Species, predict(ldatrain, newdata=Xtest)$class)
+     confusionTest
+     n <- sum(confusionTest)
+     aer[k] <- (n - sum(diag(confusionTest))) / n
+ }
> mean(aer)
[1] 0.022
```

QDA in R via the qda Function (MASS Package)

```
# fit qda model
> library(MASS)
> qdamod <- qda(Species ~ ., data=iris, prior=rep(1/3, 3))
> names(qdamod)
 [1] "prior"   "counts"  "means"   "scaling" "ldet"    "lev"     "N"
 [8] "call"    "terms"   "xlevels"

# check the QDA coefficients/scalings
> dim(qdamod$scaling)
[1] 4 4 3
> round(crossprod(qdamod$scaling[,,1], cov(Xs[,1:p])) %*% qdamod$scaling[,,1], 4)
  1 2 3 4
1 1 0 0 0
2 0 1 0 0
3 0 0 1 0
4 0 0 0 1
> round(crossprod(qdamod$scaling[,,2], cov(Xc[,1:p])) %*% qdamod$scaling[,,2], 4)
  1 2 3 4
1 1 0 0 0
2 0 1 0 0
3 0 0 1 0
4 0 0 0 1
> round(crossprod(qdamod$scaling[,,3], cov(Xv[,1:p])) %*% qdamod$scaling[,,3], 4)
  1 2 3 4
1 1 0 0 0
2 0 1 0 0
3 0 0 1 0
4 0 0 0 1
```

Plot QDA Results: All Pairwise Partitions




```
library(klaR)
species <- factor(rep(c("s","c","v"), each=50))
partimat(x=iris[,1:4], grouping=species, method="qda")
```

APER and Expected AER

```
# make confusion matrix (and APER)
> confusion <- table(iris$Species, predict(qdamod)$class)
> confusion

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         48         2
  virginica       0          1        49
> n <- sum(confusion)
> aper <- (n - sum(diag(confusion))) / n
> aper
[1] 0.02

# use CV to get expected AER
> qdamodCV <- qda(Species ~ ., data=iris, prior=rep(1/3, 3), CV=TRUE)
> confusionCV <- table(iris$Species, qdamodCV$class)
> confusionCV

            setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         3
  virginica       0          1        49
> eaer <- (n - sum(diag(confusionCV))) / n
> eaer
[1] 0.02666667
```

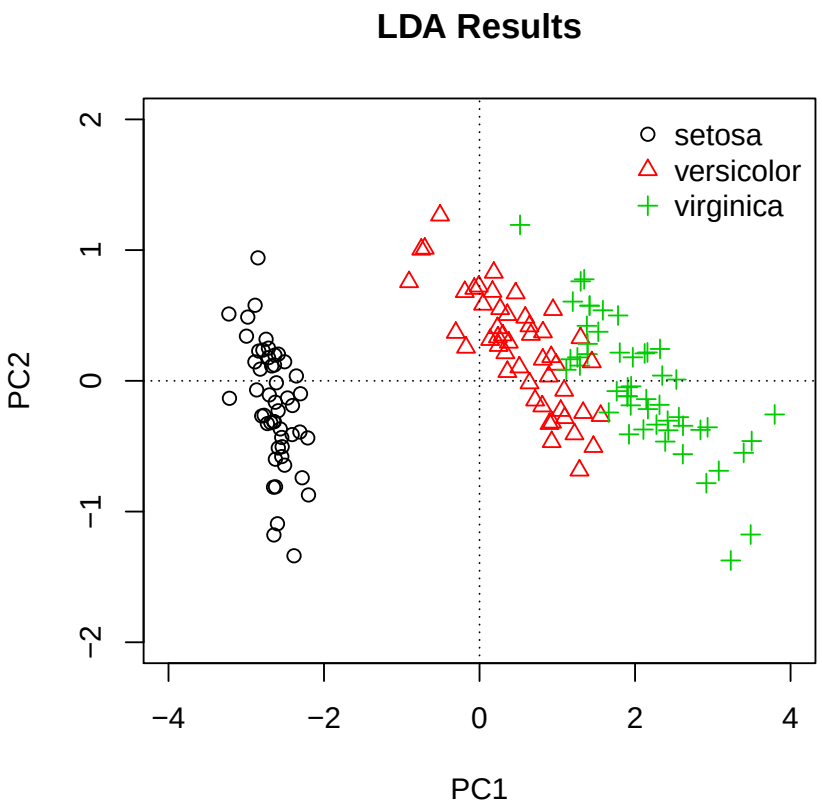
Split Data into Training (70%) and Testing (30%) Sets

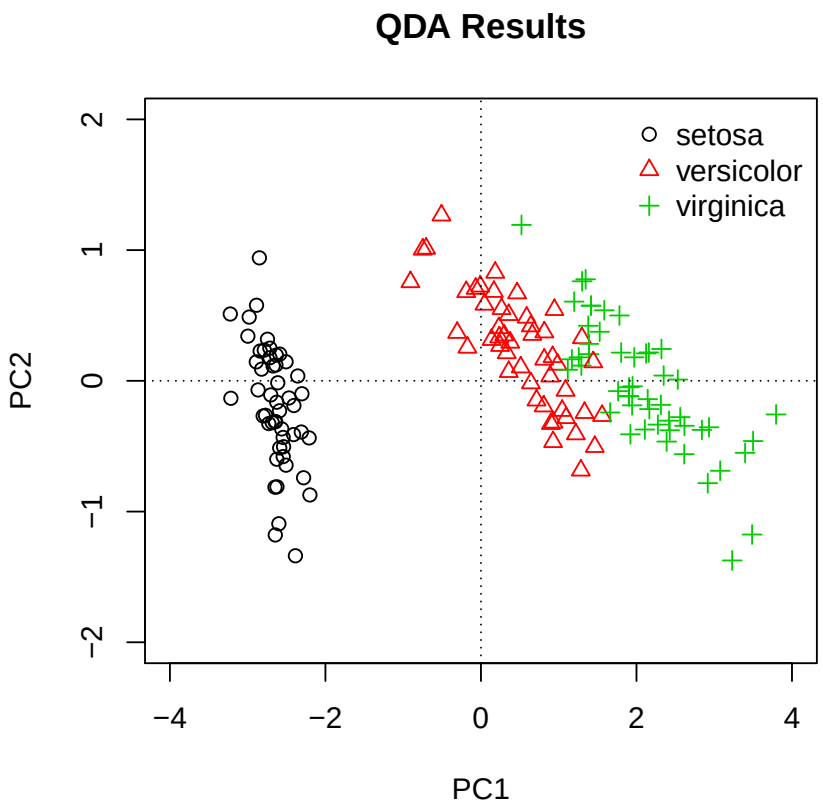
```
> # split into separate matrices for each flower
> Xs <- subset(iris, Species=="setosa")
> Xc <- subset(iris, Species=="versicolor")
> Xv <- subset(iris, Species=="virginica")

> # split into training and testing
> set.seed(1)
> sid <- sample.int(n=50, size=35)
> cid <- sample.int(n=50, size=35)
> vid <- sample.int(n=50, size=35)
> Xtrain <- rbind(Xs[sid,], Xc[cid,], Xv[vid,])
> Xtest <- rbind(Xs[-sid,], Xc[-cid,], Xv[-vid,])

# fit qda to training and evaluate on testing
> qdatrain <- qda(Species ~ ., data=Xtrain, prior=rep(1/3, 3))
> confusionTest <- table(Xtest$Species, predict(qdatrain, newdata=Xtest)$class)
> confusionTest

            setosa versicolor virginica
  setosa        15          0         0
  versicolor     0         15         0
  virginica      0          1        14
> n <- sum(confusionTest)
> aer <- (n - sum(diag(confusionTest))) / n
> aer
[1] 0.02222222
```

Two-Fold CV with 100 Random 70/30 Splits

```
> nrep <- 100
> aer <- rep(0, nrep)
> set.seed(1)
> for(k in 1:nrep){
+     sid <- sample.int(n=50, size=35)
+     cid <- sample.int(n=50, size=35)
+     vid <- sample.int(n=50, size=35)
+     Xtrain <- rbind(Xs[sid,], Xc[cid,], Xv[vid,])
+     Xtest <- rbind(Xs[-sid,], Xc[-cid,], Xv[-vid,])
+     qdatrain <- qda(Species ~ ., data=Xtrain, prior=rep(1/3, 3))
+     confusionTest <- table(Xtest$Species, predict(qdatrain, newdata=Xtest)$class)
+     confusionTest
+     n <- sum(confusionTest)
+     aer[k] <- (n - sum(diag(confusionTest))) / n
+ }
> mean(aer)
[1] 0.02466667
```

Plot LDA and QDA Results using PCA





Plot LDA and QDA Results using PCA (R code)

R code for plot on previous slide:

```
# visualize LDA and QDA results via PCA
ldaid <- as.integer(predict(ldamod)$class)
qdaid <- as.integer(predict(qdamod)$class)
pcamod <- princomp(iris[,1:4])
dev.new(width=10, height=5, noRStudioGD=TRUE)
par(mfrow=c(1,2))
plot(pcamod$scores[,1:2], xlab="PC1", ylab="PC2", pch=ldaid, col=ldaid,
     main="LDA Results", xlim=c(-4, 4), ylim=c(-2, 2))
legend("topright",lev,pch=1:3,col=1:3,bty="n")
abline(h=0,lty=3)
abline(v=0,lty=3)
plot(pcamod$scores[,1:2], xlab="PC1", ylab="PC2", pch=qdaid, col=qdaid,
     main="QDA Results", xlim=c(-4, 4), ylim=c(-2, 2))
legend("topright",lev,pch=1:3,col=1:3,bty="n")
abline(h=0,lty=3)
abline(v=0,lty=3)
```