# Classification II: Discriminant Analysis

## -Applied Multivariate Analysis-

Lecturer: Darren Homrighausen, PhD

# Classification

Logistic regression, which is the main type of GLM we are considering, directly models

$$\pi(x) = Pr(Y = 1|X = x)$$

using the logistic function.

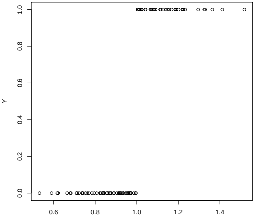
There is an alternate approach that models the distribution of the $X's$ directly and then inverts the probability via Bayes' theorem.

# Why Would we Want to do That?

There are several drawbacks to logistic regression:

- If the classes are well-separated, logistic regression is unstable (or undefined)
- It is awkward to use when the response has multiple levels

# Example of Well Separated Classes:



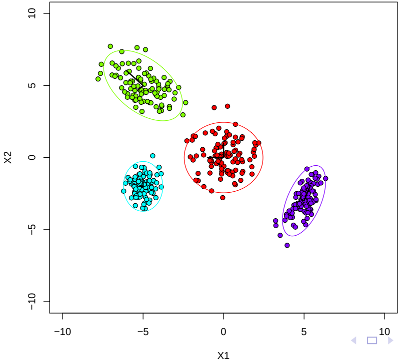

```
> glm(Y~X,family='binomial')
(Intercept)            X
     -986.2        974.2
Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:      138.3
Residual Deviance: 1.989e-08  AIC: 4
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

# What is a Gaussian?

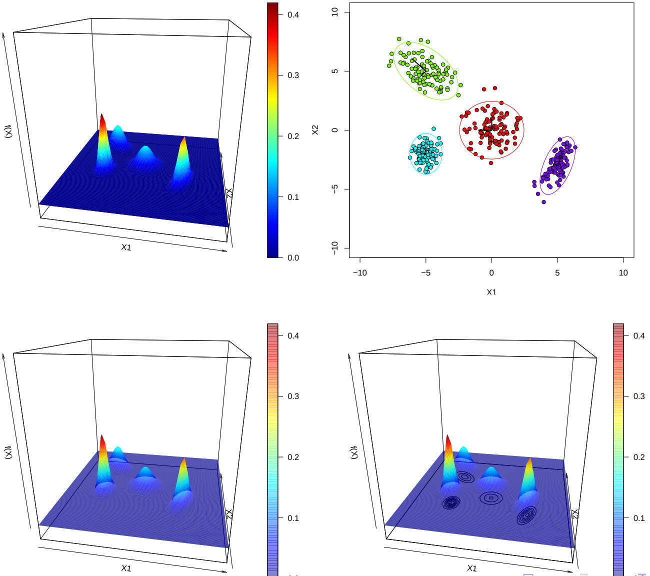
Suppose

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \sim N\left(\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \mathrm{var}(X_1) & \mathrm{cov}(X_1, X_2) \\ \mathrm{cov}(X_2, X_1) & \mathrm{var}(X_2) \end{bmatrix}\right)$$

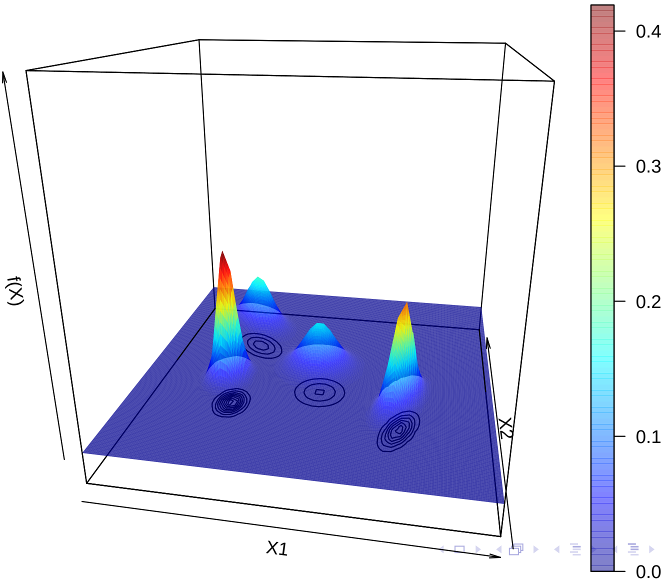
Here are $n = 100$ draws from four different Gaussian distributions.



# What is a Gaussian?



# What is a Gaussian?



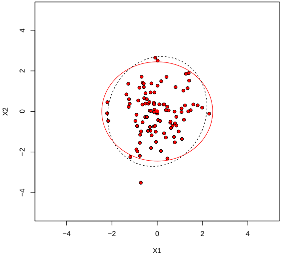
# Estimate $\mu$ and $\Sigma$?

Suppose we make $n = 100$ independent observations

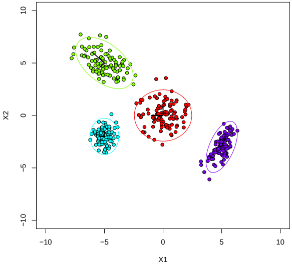
$$X_1, \dots, X_{100} \sim N\left(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$$

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i = \begin{bmatrix} 0.0012 \\ 0.001 \end{bmatrix}$$

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \overline{X})(X_i - \overline{X})^\top = \frac{1}{n} (\mathbb{X} - \overline{\mathbb{X}})^\top (\mathbb{X} - \overline{\mathbb{X}}) = \begin{bmatrix} 0.8 & 0.1 \\ 0.1 & 1.2 \end{bmatrix}$$



# ESTIMATING $\mu$ AND $\Sigma$ WITH SEVERAL GAUSSIANS




Suppose we want to estimate different Gaussians at the same time

Let $k = 1, \ldots, K$ index these groups
($K = 4$ in figure)

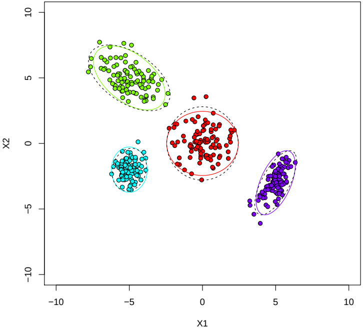
- $X_{1k}, \ldots, X_{n_k k}$ be from group $k$
- $n_k$ be the number of observations in $k^{th}$ group
- $n = \sum_{k=1}^{K} n_k$

# Estimating several different Gaussians

We can estimate these groups with

$$\overline{X}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} X_{ik}$$

$$\hat{\Sigma}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (X_{ik} - \overline{X}_k)(X_{ik} - \overline{X}_k)^\top$$



# Estimating several different Gaussians

A problem with this approach: a lot of parameters

Each covariance matrix has: $p(p+1)/2$ parameters

(As $\hat{\Sigma}_k$ must be symmetric)

For $K$ groups, this means $Kp(p+1)/2$ parameters

For this problem:

$$Kp(p+1)/2 = 12$$

This can be very large for even moderately large $p$ or $K$

For $p = 50$:

$$Kp(p+1)/2 = 5100$$

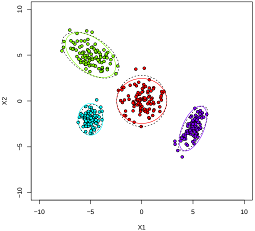
# How to estimate $\mu$ and $\Sigma$ with a mixture of Gaussians

There isn't much we can do about the $p(p+1)/2$ part

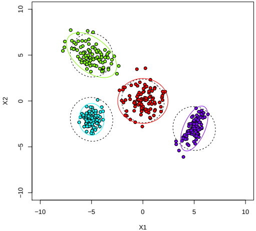
But, we can make this simplification: Assume $\Sigma_k = \Sigma$

(This means we use all observations to estimate a single covariance)

$$\hat{\Sigma} = \frac{1}{n} \sum_{k=1}^{K} \sum_{i=1}^{n_k} (X_{ik} - \overline{X}_k)(X_{ik} - \overline{X}_k)^\top$$




Different $\hat{\Sigma}_k$




All same $\hat{\Sigma}$

# Linear Discriminant Analysis

# Linear Discriminant Analysis (LDA)

Suppose our response can take on $K$ different levels:

$$Y = \begin{cases} 1 \\ \vdots \\ K \end{cases}$$

1. We model the covariates as a Gaussian random variable
   ($X | Y = k \sim N(\mu_k, \Sigma)$)
2. Specify prior probabilities of that $Y = k$
   ($\pi_k = \mathbb{P}(Y = k)$)
3. Turn this into a conditional distribution of $Y$ given $X$
   (Using Bayes' theorem)
4. Find the best possible classifier
   (This is the Bayes' rule)
5. This depends on the unknown parameters
   $\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma$.
6. Estimate these parameters with their sample versions.

# What is Bayes' theorem?

Here, we are interested in the class label $Y = k$ at particular covariate value $X$

That is, we want

$$\mathbb{P}(Y = k|X)$$

(Recall, this is the main ingredient to the Bayes' rule)

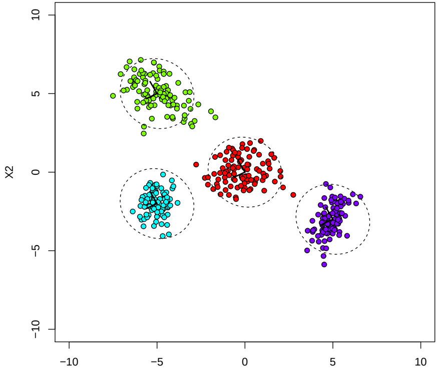
Bayes' theorem:

$$\mathbb{P}(Y = k|X) = \frac{\mathbb{P}(X|Y = k)\mathbb{P}(Y = k)}{\mathbb{P}(X)}$$

- $\mathbb{P}(X|Y = k) = N(\mu_k, \Sigma)$
- $\mathbb{P}(Y = k) = \pi_k$

# Intuition

How would you classify a point with this data?

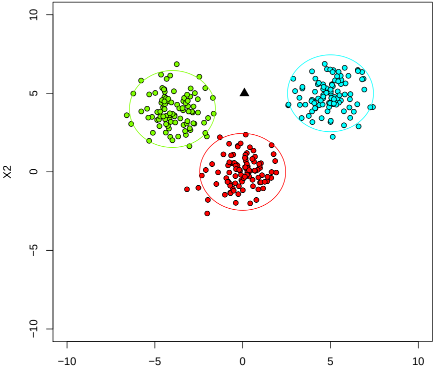



Effectively we just classify an observation to the closest mean ($\overline{X}_k$)

What do we mean by close? (Need to define distance)

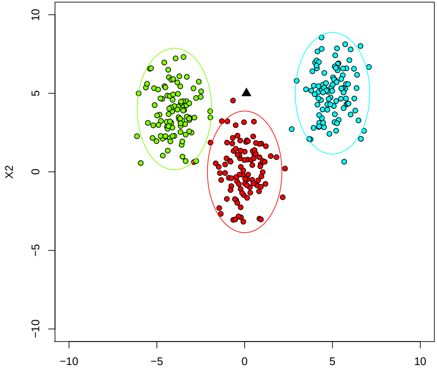
# INTUITION

What if the data looked like this?



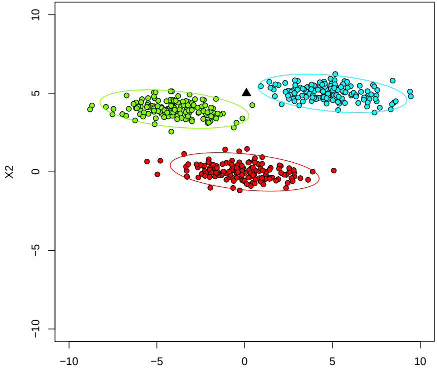
# Intuition

Or this?



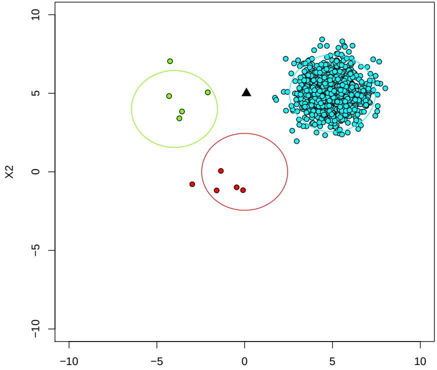
# Intuition

How about this?



# Intuition

What about now?



# Intuition

All of these examples show that we need to take into account

- The shape of distribution (size and eccentricity of the ellipse)
- The relative number of points in each group

These are the two main ingredients in LDA

# Linear discriminant analysis (LDA)

We use the linear discriminant function

$$\delta_k(\mathbf{x}) = \underbrace{\mathbf{x}^\top \hat{\Sigma}^{-1} \overline{X}_k - \frac{1}{2} \overline{X}_k^\top \hat{\Sigma}^{-1} \overline{X}_k}_{\textit{likelihood}} + \underbrace{\log(\hat{\pi}_k)}_{\textit{prior}}$$

Here, $\hat{\pi}_k$ is the fraction of observations in group $k$ (that is, $\frac{n_k}{n}$)

We assign an observation to $\hat{k}$, where

$$\hat{k} = \arg\max_k \delta_k(x)$$

# Linear discriminant analysis (LDA)

Intuitively, assigning observations to the nearest $\overline{X}_k$ (but ignoring the covariance) would amount to

$$\begin{aligned}\tilde{k} &= \underset{k}{\operatorname{argmin}} \|\mathbf{x} - \overline{X}_k\|_2^2 \\ &= \underset{k}{\operatorname{argmin}}\, \mathbf{x}^\top \mathbf{x} - 2\mathbf{x}^\top \overline{X}_k + \overline{X}_k^\top \overline{X}_k \\ &= \underset{k}{\operatorname{argmin}} -\mathbf{x}^\top \overline{X}_k + \frac{1}{2}\overline{X}_k^\top \overline{X}_k\end{aligned}$$

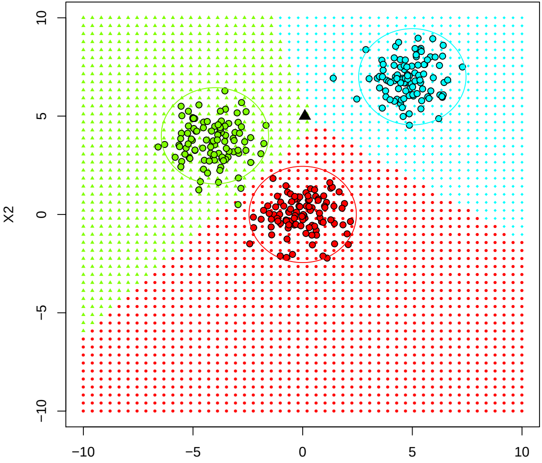
compare this to:

$$\hat{k} = \underset{k}{\arg\max} \underbrace{\mathbf{x}^\top \hat{\Sigma}^{-1} \overline{X}_k - \frac{1}{2}\overline{X}_k^\top \hat{\Sigma}^{-1} \overline{X}_k}_{\textit{likelihood}} + \underbrace{\log(\hat{\pi}_k)}_{\textit{prior}}$$

The difference is we weight the distance by $\hat{\Sigma}^{-1}$ and weight the class assignment by fraction of observations in each class.

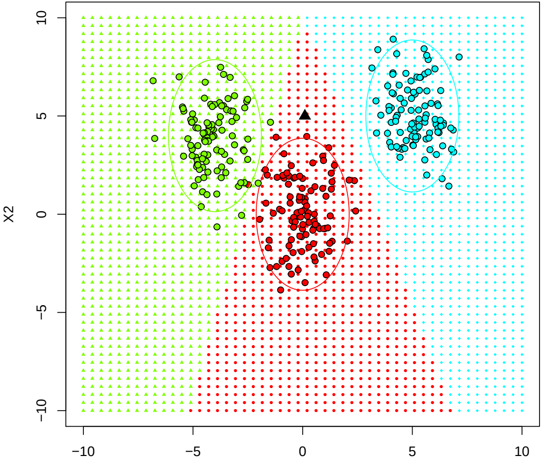
# Intuition

What if the data looked like this?



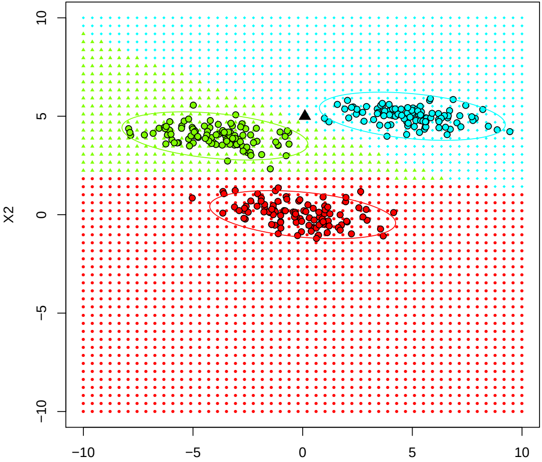
# Intuition

Or this?



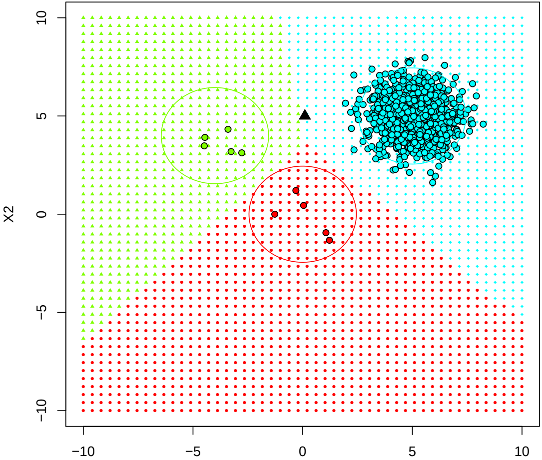
# Intuition

How about this?



# Intuition

What about now?



# LDA in R

We can do this readily in R

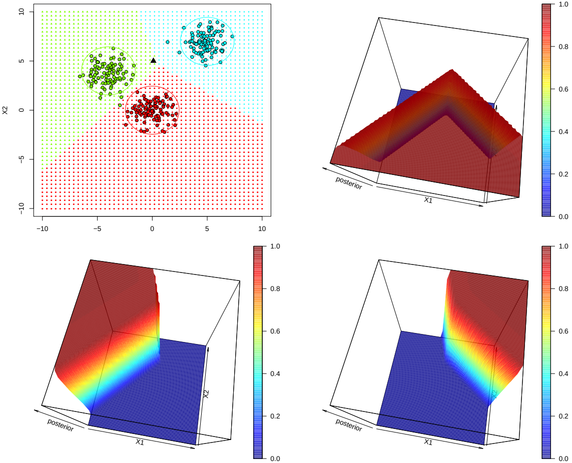
```
library(MASS)
lda.fit  = lda(Y~.,data=X)

> names(lda.fit)
 [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"

out = predict(lda.fit,X_0)

> out$posterior[1:3,]
          1            2            3
1 0.9999908 9.215567e-06 1.504633e-55
2 0.9999977 2.341924e-06 1.664446e-54
3 0.9999994 5.951430e-07 1.841223e-53
```

# What does posterior mean?




```
> print(predict(lda.fit,X_0)$posterior)
           1         2           3
1 0.04883796 0.9477494 0.003412639
```

# Recap

Reminder: For every problem, we can define: $\underset{\hat{Y}}{\operatorname{argmin}}\, \mathbb{P}(\hat{Y} \neq Y_0)$

This is known as the Bayes' rule

It looks like (for $Y$ taking either 0 or 1):

$$0 \text{ if } \mathbb{P}(Y = 0|X) \geq \mathbb{P}(Y = 1|X)$$

or

$$1 \text{ if } \mathbb{P}(Y = 1|X) \geq \mathbb{P}(Y = 0|X)$$

(That is, we want to maximize the conditional probability)

Emphasis: The Bayes' rule is unknown/unknowable

With LDA we are trying to estimate it under particular assumptions
(Concept check: What are the assumptions?)

# Performance of LDA

The quality of the classifier produced by LDA depends on two things:

- The sample size $n$

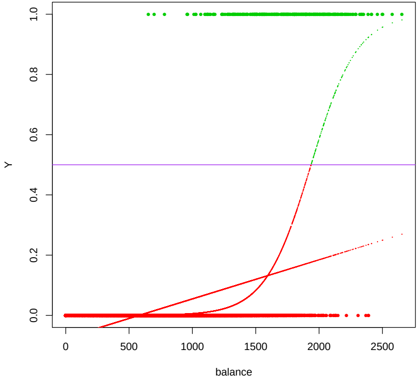
  (This determines how accurate the $\hat{\pi}_k$, $\hat{\mu}_k$, and $\hat{\Sigma}$ are)
- How wrong the LDA assumptions are

  (That is: $X|Y = k$ is a Gaussian with mean $\mu_k$ and variance $\Sigma$)

Recall: The decision boundary of a classifier are the values of $X$ such that the classifier is indifferent between two (or more) levels of $Y$

A linear decision boundary is when this set of values looks like a line

# We've Already Seen Other Examples of Linear Decision Boundaries



# LDA: under correct assumptions

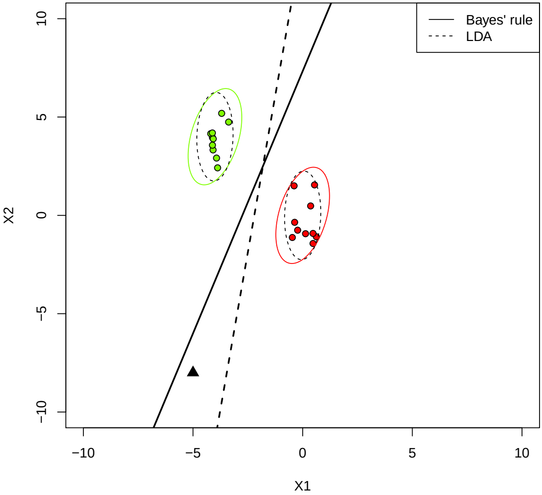



Figure: For $n_k = 10$

# LDA: under correct assumptions

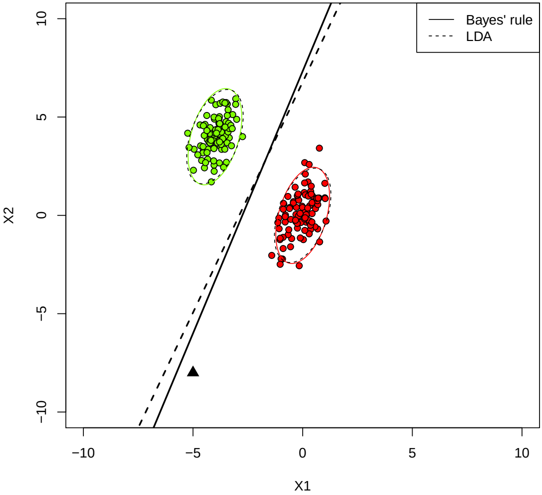



FIGURE: For $n_k = 100$

# LDA: under correct assumptions

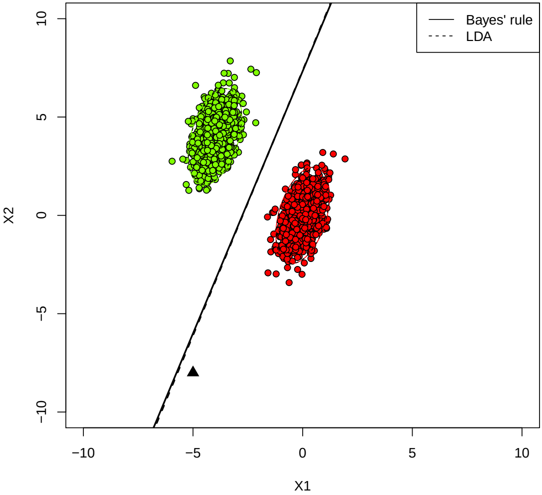



FIGURE: For $n_k = 1000$

# LDA: mildly incorrect assumptions

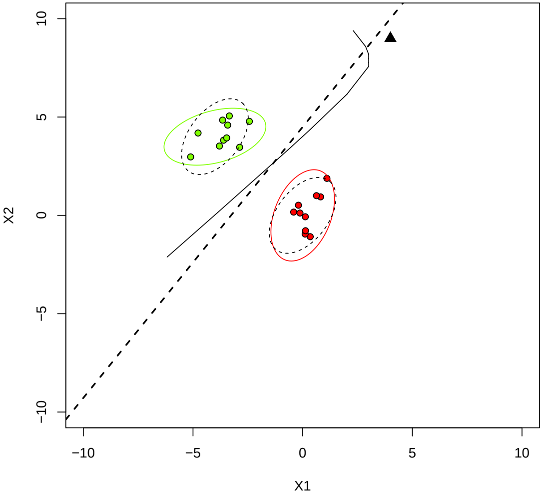



Figure: For $n_k = 10$

# LDA: mildly incorrect assumptions

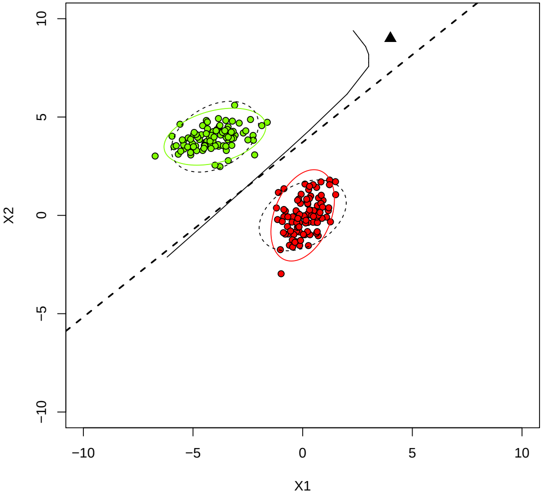



Figure: For $n_k = 100$

# LDA: mildly incorrect assumptions

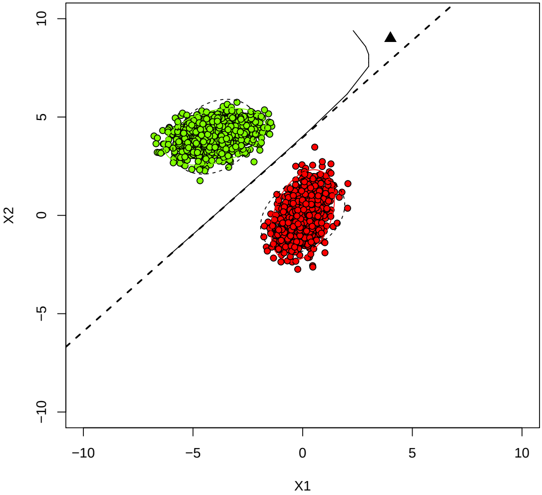



Figure: For $n_k = 1000$

# LDA: very incorrect assumptions

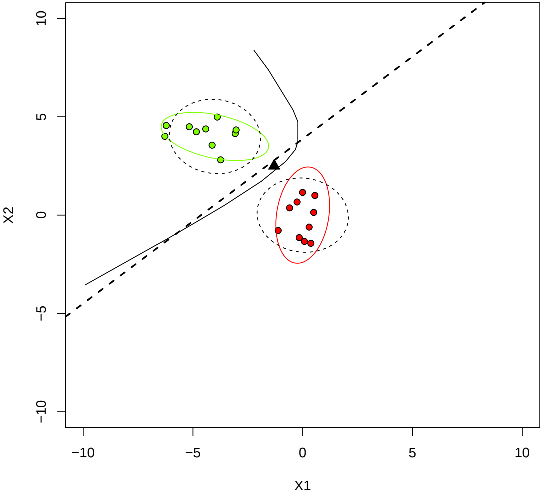



FIGURE: For $n_k = 10$

# LDA: very incorrect assumptions

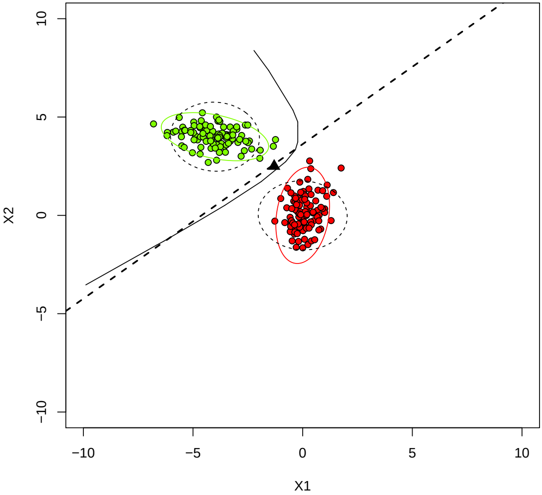



FIGURE: For $n_k = 100$

# LDA: very incorrect assumptions

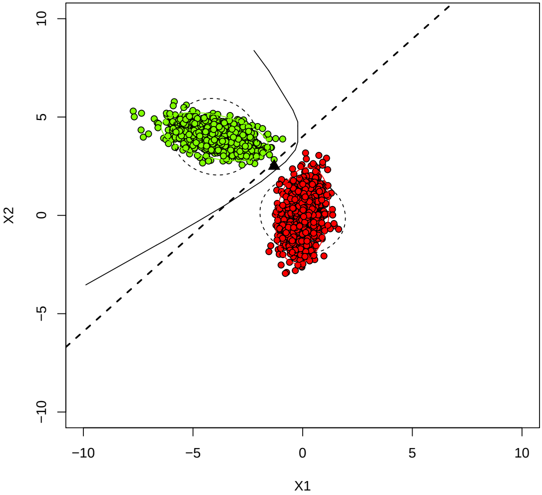


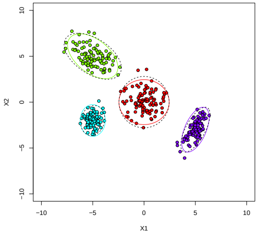

Figure: For $n_k = 1000$

# The LDA variance assumption

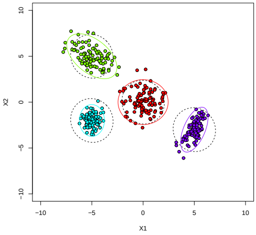
Returning to the assumption: $\Sigma_k = \Sigma$

The assumption provides two benefits:

- Allows for estimation when $n$ isn't large compared with $Kp(p+1)/2$
- Lowers the variance of the procedure (but produces bias)
  (This can be seen by the estimation of fewer parameters)




Different $\hat{\Sigma}_k$




All same $\hat{\Sigma}$

# The LDA variance assumption

However, when

- $n$ is large compared with $Kp(p+1)/2$
  (Say, $\min n_k \geq 40p(p+1)/2$)
- The induced bias outweighs the variance
  (This is hard to determine. Usually compare the prediction error on test set)

We relax the assumption and let $X|Y = k$ have

- mean $\mu_k$
- variance $\Sigma_k$

These additional parameters make the decision boundary quadratic
(Instead of linear)

# Quadratic Discriminant Analysis

# Quadratic discriminant analysis (QDA)

The formulas for QDA are a bit more complicated, so I'll omit them

However, the motivation is the same: classify with the label of the closest group, taking into account:

- The covariance of every group ($\Sigma_k$)
- The relative probability of each group ($\pi_k$)

It has almost exactly the same R code:

```
library(MASS)
qda.fit  = qda(Y~.,data=X)

out = predict(lda.fit,X_0)
```

# QDA: More flexibility than needed

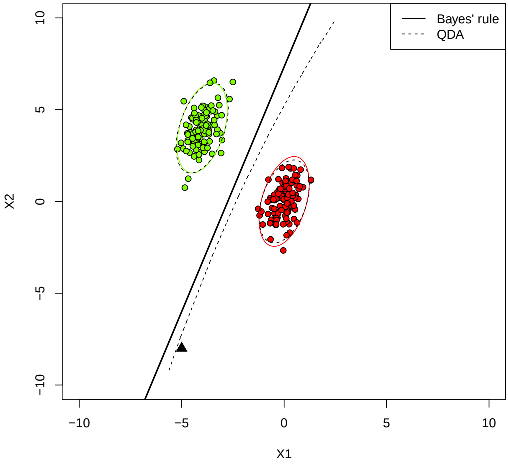



Figure: For $n_k = 100$. Note linear Bayes' rule, nonlinear QDA decision boundary

# QDA: More flexibility than needed

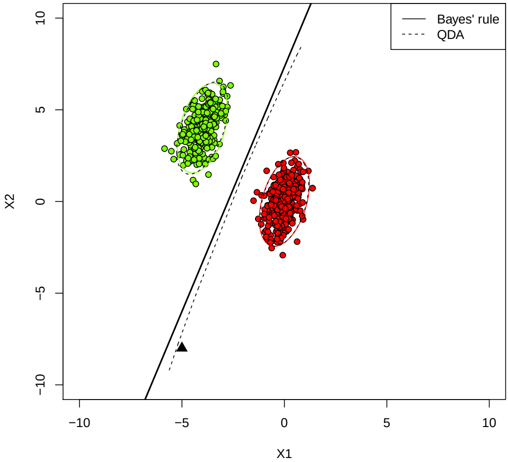



Figure: For $n_k = 300$. Note linear Bayes' rule, nonlinear QDA decision boundary

# QDA: More flexibility than needed

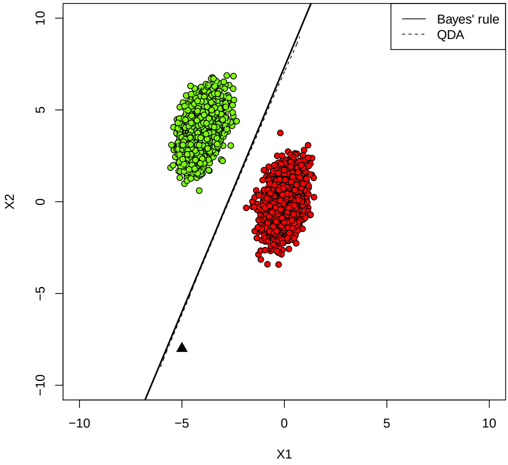



Figure: For $n_k = 2000$. Note linear Bayes' rule. The nonlinear QDA decision boundary has converged to Bayes' rule

# QDA: Different $\Sigma_k$ assumption needed

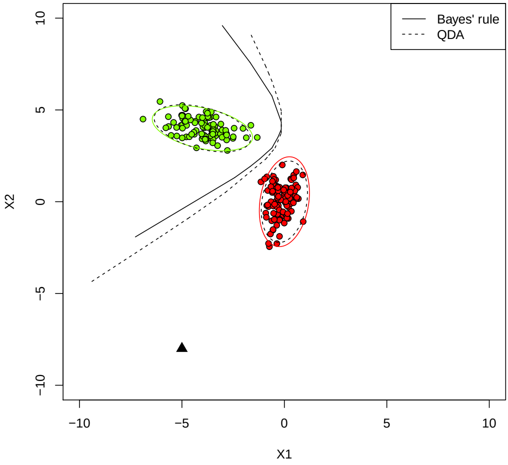



Figure: For $n_k = 100$. Note nonlinear Bayes' rule, nonlinear QDA decision boundary

# QDA: Different $\Sigma_k$ assumption needed

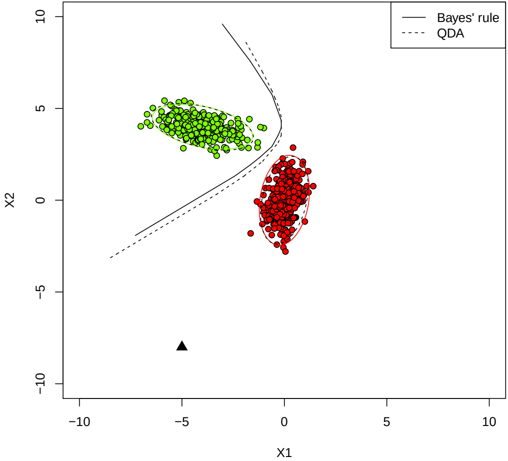



FIGURE: For $n_k = 300$. Note nonlinear Bayes' rule, nonlinear QDA decision boundary

# QDA: Different $\Sigma_k$ assumption needed

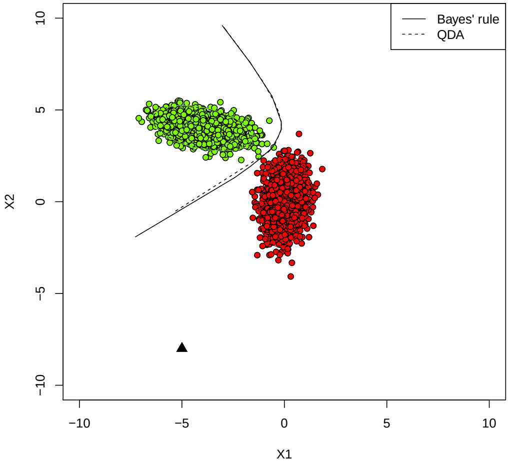



Figure: For $n_k = 2000$. Note nonlinear Bayes' rule, nonlinear QDA decision boundary

# LDA vs. QDA: under correct assumptions

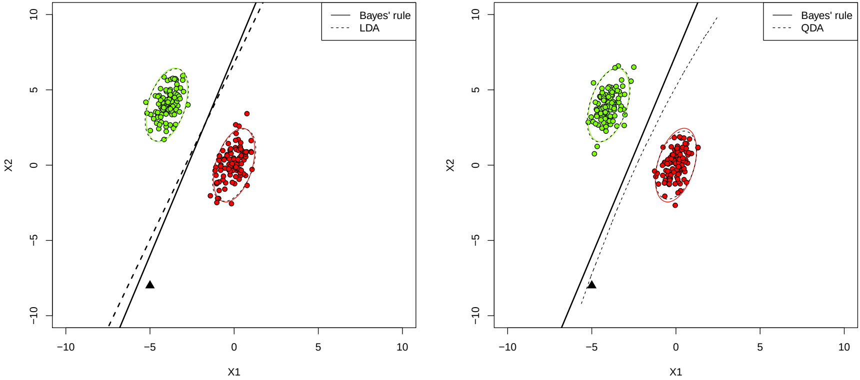



FIGURE: For $n_k = 100$

# LDA vs. QDA: very incorrect assumptions

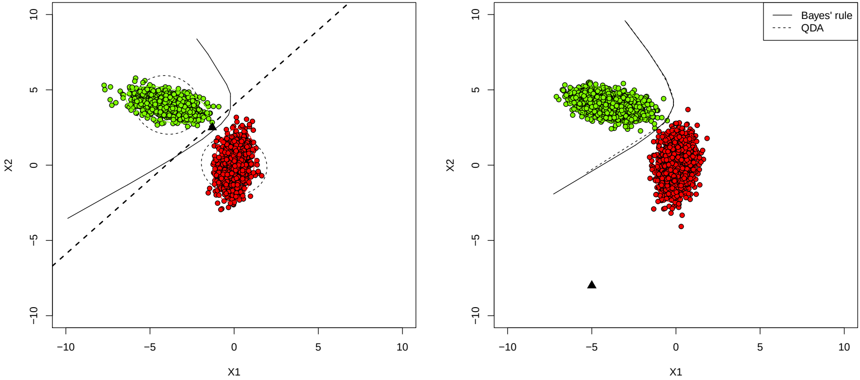



FIGURE: LDA $n_k = 1000$, QDA $n_k = 2000$