Classification: Generative Models (LDA and QDA)

Linear Discriminant Analysis

Quadratic Discriminant Analysis

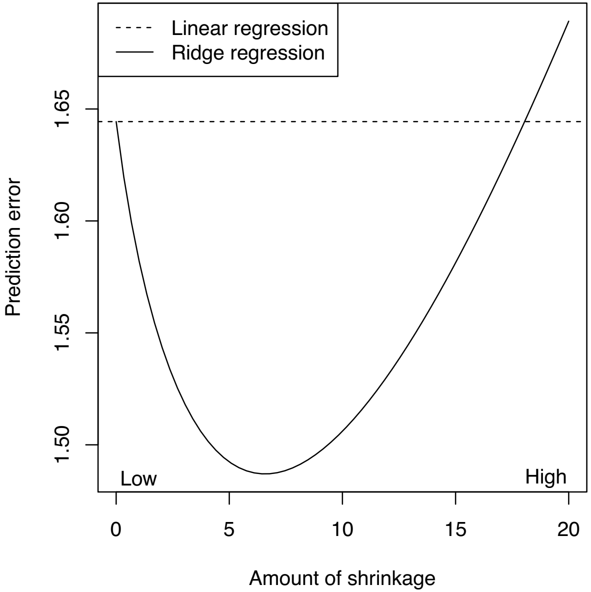
Siva Balakrishnan
Data Mining: 36-462/36-662

February 5th, 2018

Chapter 4.4 of ISL

Recap: Regularization – Why?

- Trade-off fit with model complexity for better generalization.



- Other benefits: simpler (sparse) models may be cheaper/faster to evaluate, easier to interpret

depend on fewer predictors

Recap: Regularization – How?

- Lots of different ways. Two basic, popular ones.
- Ridge penalty:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

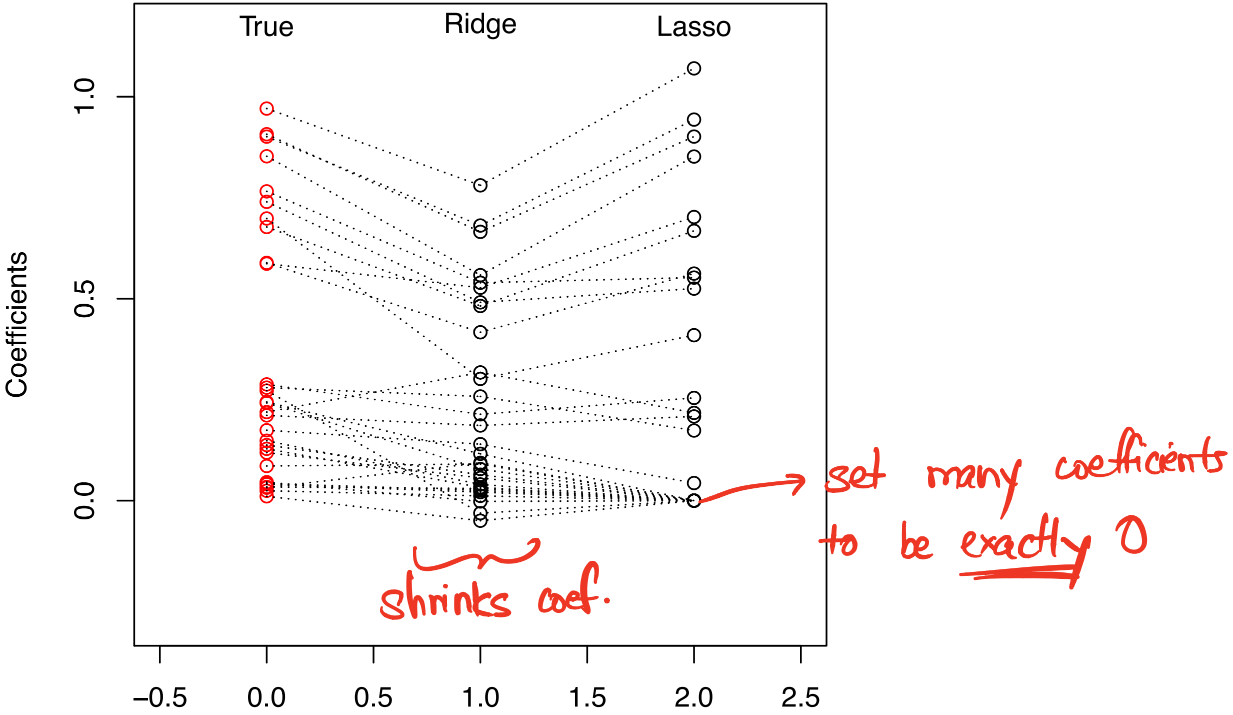
- LASSO penalty:

$$\hat{\beta}_{LASSO} = \arg\min_{\beta} \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^T\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

- Roughly, if we care solely about prediction, and suspect that most of our predictors are useful (perhaps have moderately large coefficients) use Ridge. If you have a very high-dimensional problem (many predictors), or if you care about more than just prediction (perhaps interpretability) then use LASSO.

Recap: Regularization – in a picture

Our running example from last time: $n = 50$, $p = 30$, $\sigma^2 = 1$, 10 large true coefficients, 20 small. Here is a visual representation of LASSO vs. ridge coefficients (with the same degrees of freedom):



Assumptions Philosophy

- You might be used to seeing for example the method of linear regression motivated by a series of assumptions:
  - $\underline{\mathbb{E}[y|X = x] = \beta^T x}$ or $\underbrace{y = \beta^T x + \epsilon}$, $\underline{\mathbb{E}[\epsilon] = 0}$, or $\underbrace{\epsilon \sim N(0, \sigma^2)}$
  - $\epsilon \perp\!\!\!\perp X$
  - Predictors are not highly correlated.
  - Observations are i.i.d.
  - $\vdots$

to motivate using least squares!

- None of these assumptions are necessary to use the method of (regularized) least squares, and in general the method might yield useful predictions even when none of these assumptions are true (or when they are true in some approximate sense).
- "ML/Statistical Learning Philosophy": Focus mainly on prediction (less on inference), evaluate using held-out-set performance.

Weak and Strong Modeling

- It is worth thinking about stages of method development:
  1. Constructing the method
  2. Evaluating or interpreting the output of the method
  3. Studying the properties of the method
- *Strong Modeling:* Make strong assumptions about the data through stages (1)-(3).
- *Weak Modeling:* Make strong assumptions about the model for stage (1). Do everything else more pragmatically.
- I personally like the weak modeling approach.

Another Example

- In classification with logistic regression: we have implicitly made many modelling assumptions:

  $\mathbb{E}[y|X=x] = \frac{\exp(\beta^T x)}{1+\exp(\beta^T x)}$

  - That the conditional distribution of $y|X$ has a certain form
  - That the log-likelihood is a reasonable way to fit the model, the data is i.i.d.

    not obvious why.
  - That the loss we care about is the 0/1 loss
  - $\vdots$
- Our preference for weak modeling dictates that we use our modeling assumptions to derive the method (we did this) but then we move away from assumptions when evaluating the method. We simply evaluate our classifier using held-out data (relatively assumption-light).
- If it does not work, we try to understand why (was log-likelihood a bad choice, was 0/1 loss inadequate, was a linear decision boundary an over-simplification, is it bias or variance that is hurting us, ...) and use this to identify better methods.

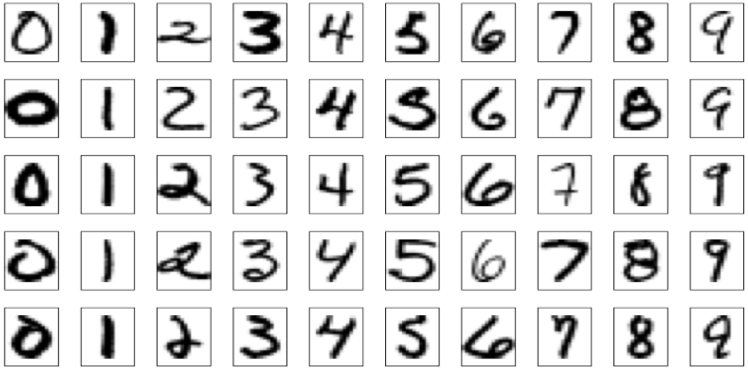
  maybe data was imbalanced

  are you overfitting or underfitting?

Changing Gears: Back to Generative Classifiers

Generative models are nice, because they let us think about modeling the process that *generated the data*

Think about the classic MNIST digits data:



It might be easier to describe what a 4 looks like than how all the digits are different.

Why do we need another classifier?

- Generative classifiers allow us to model problem in a different way.
- Logistic regression can be unstable for linearly separable data. More generally, need to regularize.
- Every classification method is derived from *different assumptions* about the problem. When these assumptions are true/close to true then the corresponding classifier will do very well.
- So for instance, Linear Discriminant Analysis (LDA) will likely outperform logistic regression when the conditional distribution of $X|y = k$ are Gaussian with the same covariance matrix.

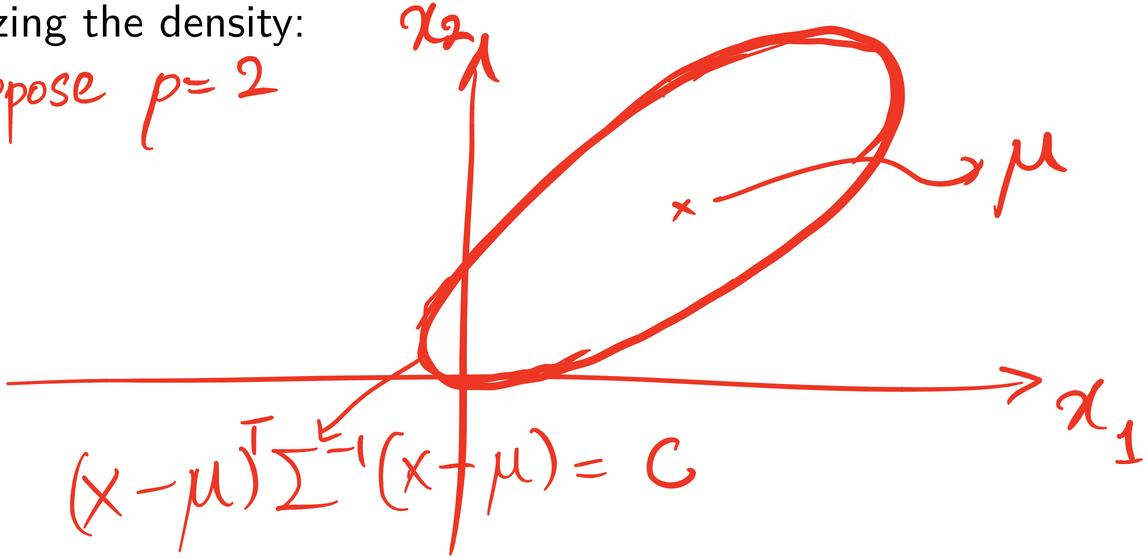
Reminder: Multivariate Gaussian

- Density: $x \in \mathbb{R}^p$ $\quad x \sim N(\overset{\in \mathbb{R}^p}{\mu}, \underset{\in \mathbb{R}^{p \times p}}{\Sigma})$

$$p(x) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp\left\{ \frac{-(x-\mu)^T \Sigma^{-1} (x-\mu)}{2} \right\}$$

- Visualizing the density:

Suppose $p = 2$



$$(x-\mu)^T \Sigma^{-1} (x-\mu) = c$$

Reminder: Multivariate Gaussian, Maximum Likelihood

- Suppose we observed $X_1, \ldots, X_n$ and wanted to fit a Gaussian to this data. → need to estimate (μ, Σ).
- **Maximum Likelihood Mean Estimation:**

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

- **Maximum Likelihood Covariance Matrix Estimation:**

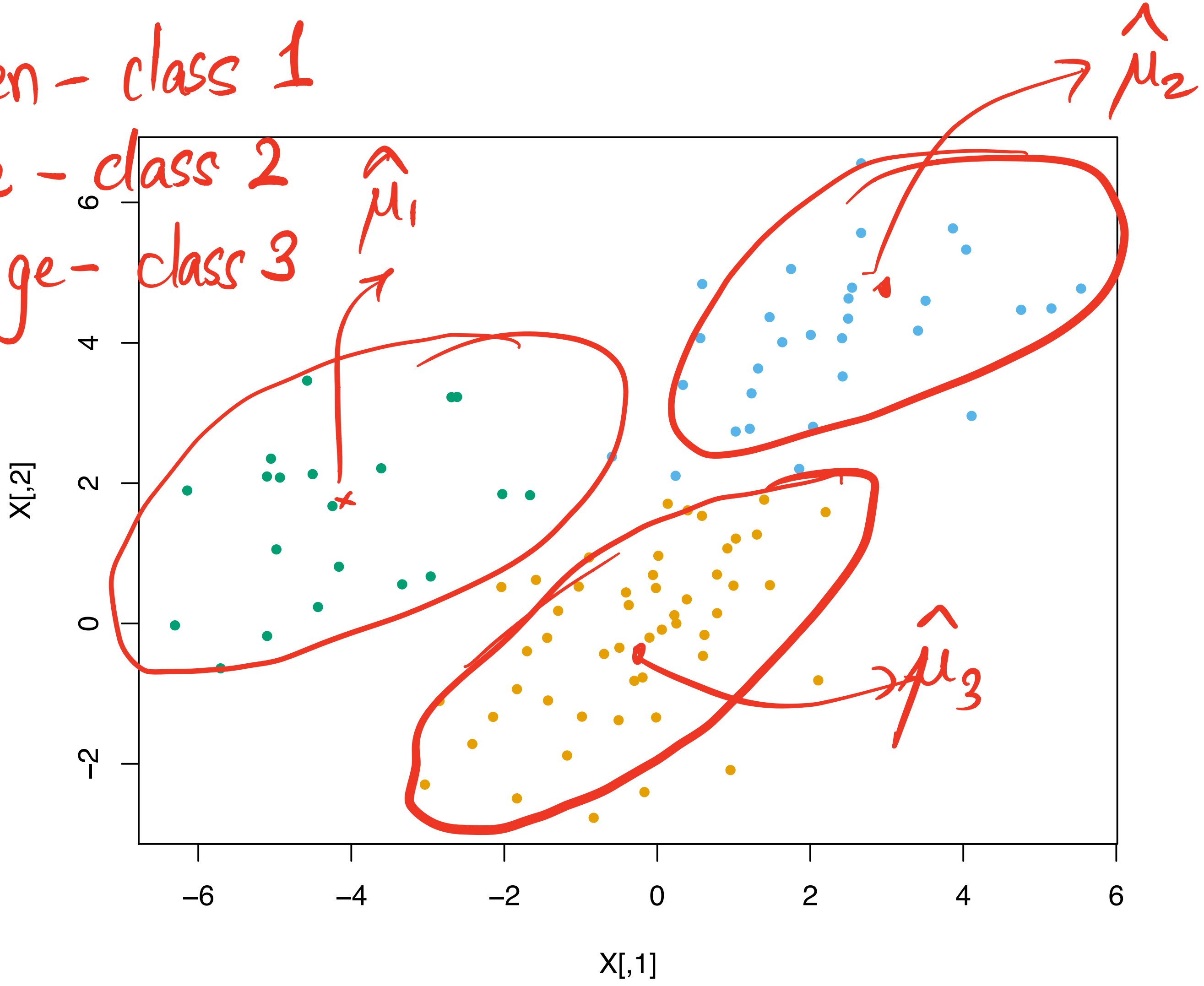
$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T$$

can also use: $\hat{\Sigma}_u = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^T$

Green - class 1

Blue - class 2

Orange - class 3



Basic Idea of Generative Classifiers

- Suppose that we knew $f_k(x) = \mathbb{P}(X = x|Y = k)$ for all of our classes k. f_k is the density function for a particular x for a particular class.
- Roughly, we would think it is unlikely that a particular observation came from class k if $f_k(x)$ is very small.
- More precisely, we can use Bayes theorem:

$$\mathbb{P}(Y=k|X=x) = \frac{\mathbb{P}(Y=k) f_k(x)}{\sum_j \mathbb{P}(Y=j) f_j(x)}$$

posterior probability

$$\mathbb{P}(Y=k) := \pi_k$$

~ prior probabilities

Linear discriminant analysis

We need to estimate $\mathrm{P}(X = x|Y = j)$ and prior probabilities π_j in order to use the generative form of the Bayes classifier.

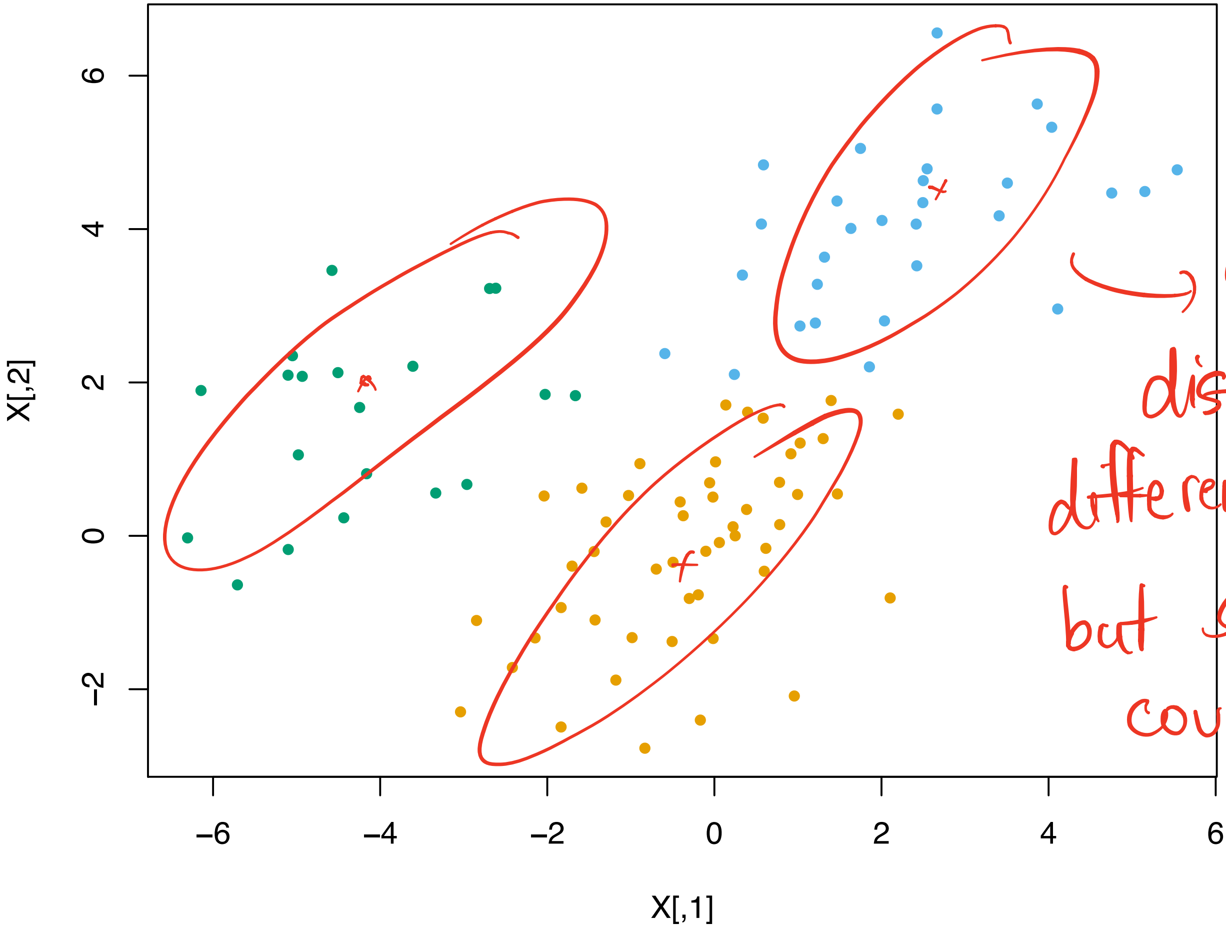
Linear discriminant analysis (LDA) does this by assuming that the data within each class are normally distributed:

$$\cancel{h}f_j(x) = \mathrm{P}(X = x|Y = j) = N(\mu_j, \Sigma) \text{ density}$$

We allow each class to have its own mean $\mu_j \in \mathbb{R}^p$, but we assume a common covariance matrix $\Sigma \in \mathbb{R}^{p \times p}$. Hence

$$\cancel{h}f_j(x) = \frac{1}{(2\pi)^{p/2} \det(\Sigma)^{1/2}} \exp\left\{ -\frac{1}{2}(x - \mu_j)^T \Sigma^{-1} (x - \mu_j) \right\}$$

Think of this as a model for classification problems where the different classes look like shifted clumps.



each dist. has different mean but same covariance

What does the decision rule look like? We want to find j so that $\mathrm{P}(Y=j|X=x)\cdot\pi_j = f_j(x)\cdot\pi_j$ is the largest. We can define the rule:

$$\begin{aligned}
f^{\mathrm{LDA}}(x) &= \underset{j=1,\ldots K}{\operatorname{argmax}} \; \frac{1}{(2\pi)^{p/2}\det(\Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)} \cdot \pi_j \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \downarrow \\
&= \underset{j=1,\ldots K}{\operatorname{argmax}} \; \log\Big(\qquad\qquad\qquad\qquad\qquad \Big) \\
&= \underset{j=1,\ldots K}{\operatorname{argmax}} \; -\frac{1}{2}(x-\mu_j)^T\Sigma^{-1}(x-\mu_j) + \ln\pi_j \\
&= \underset{j=1,\ldots K}{\operatorname{argmax}} \; \delta_j(x) = x^T\Sigma^{-1}\mu_j - \frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j + \ln\pi_j
\end{aligned}$$

We call $\delta_j(x)$, $j=1,\ldots K$ the discriminant functions. Note

$$\delta_j(x) = x^T\Sigma^{-1}\mu_j - \frac{1}{2}\mu_j^T\Sigma^{-1}\mu_j + \log\pi_j$$

is just an affine function of x

In practice, we have to estimate the parameters π_j, μ_j, and Σ. We estimate them based on the training data $x_i \in \mathbb{R}^p$ and $y_i \in \{1, \dots K\}$, $i = 1, \dots n$, by:

- $\widehat{\pi}_j = n_j/n$, the proportion of observations in class j
- $\widehat{\mu}_j = \frac{1}{n_j} \sum_{y_i=j} x_i$, the centroid of class j
- $\widehat{\Sigma} = \frac{1}{n\boxed{-K}} \sum_{j=1}^{K} \sum_{y_i=j} (x_i - \widehat{\mu}_j)(x_i - \widehat{\mu}_j)^T$, the pooled sample covariance matrix

unbiased est. of pooled covariance

(Here n_j is the number of points in class j)

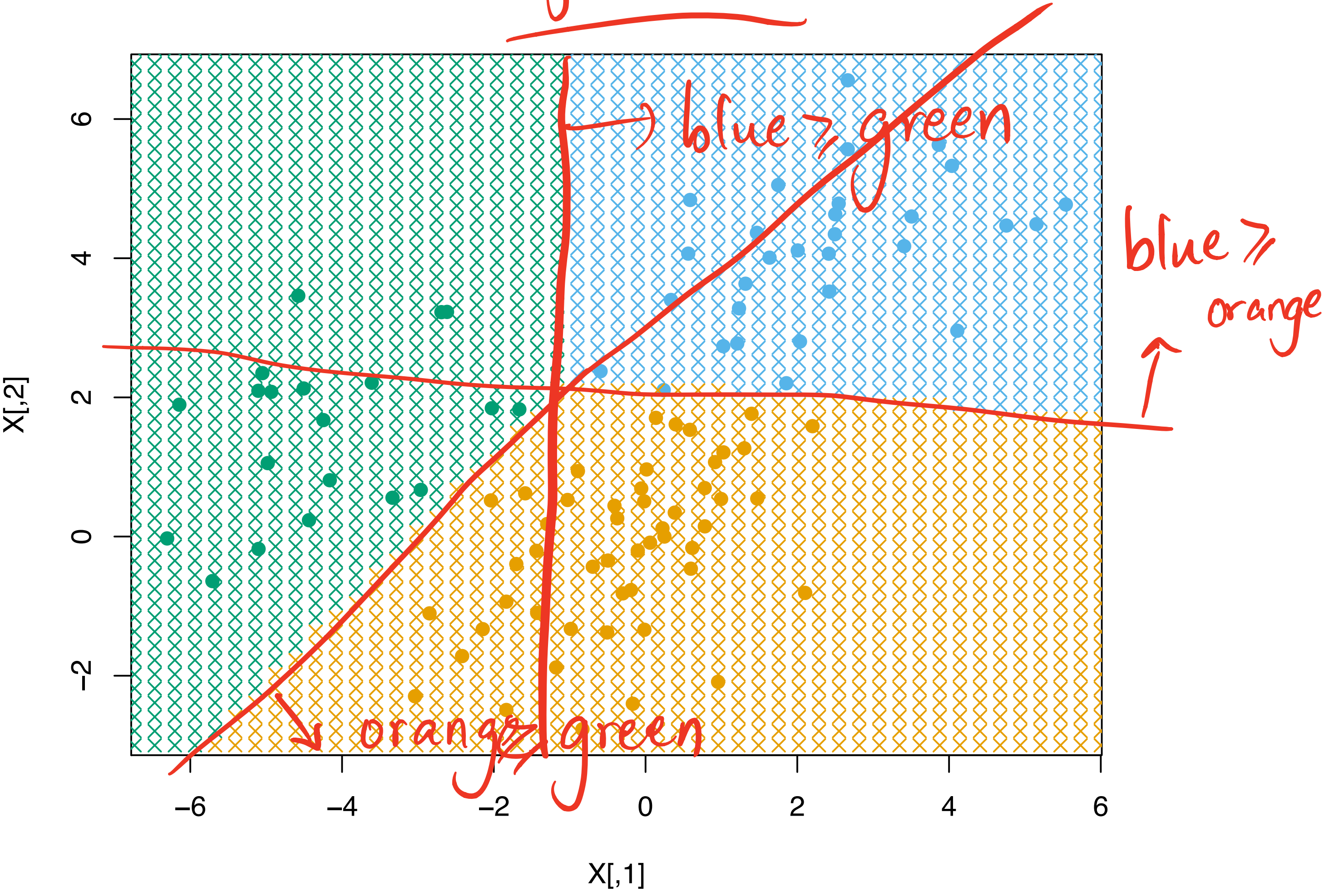
This gives the estimated discriminant functions:

$$\widehat{\delta}_j(x) = x^T \widehat{\Sigma}^{-1} \widehat{\mu}_j - \frac{1}{2} \widehat{\mu}_j^T \widehat{\Sigma}^{-1} \widehat{\mu}_j + \log \widehat{\pi}_j$$

and finally the linear discriminant analysis rule for new $x \in \mathbb{R}^p$,

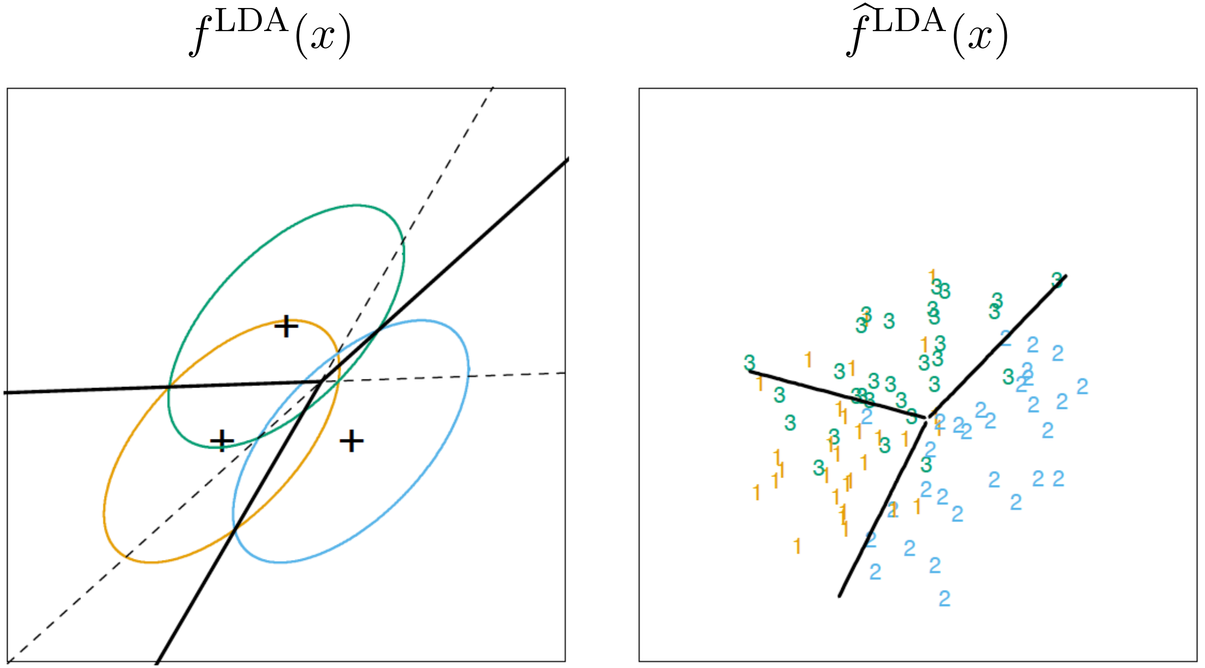
$$\widehat{f}^{\mathrm{LDA}}(x) = \underset{j=1,\dots K}{\operatorname{argmax}} \; \widehat{\delta}_j(x)$$

$\delta_j \geq \delta_k.$



Example: LDA decision boundaries

Example of decision boundaries from LDA (from ESL page 109):



$f^{\mathrm{LDA}}(x)$

$\widehat{f}^{\mathrm{LDA}}(x)$

Are the decision boundaries the same as the perpendicular bisectors between the class centroids?

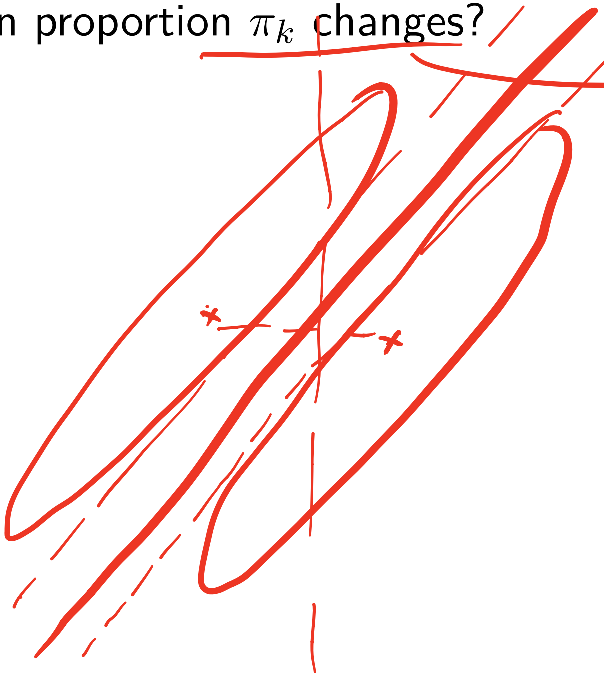
Thinking about the decision function

$$\widehat{f}^{\mathrm{LDA}}(x) = \operatorname*{argmax}_{j=1,\ldots K} x^T \widehat{\Sigma}^{-1} \widehat{\mu}_j - \frac{1}{2} \widehat{\mu}_j^T \widehat{\Sigma}^{-1} \widehat{\mu}_j + \log \widehat{\pi}_j$$

What changes if our loss function is not 0-1? What changes if the population proportion π_k changes?

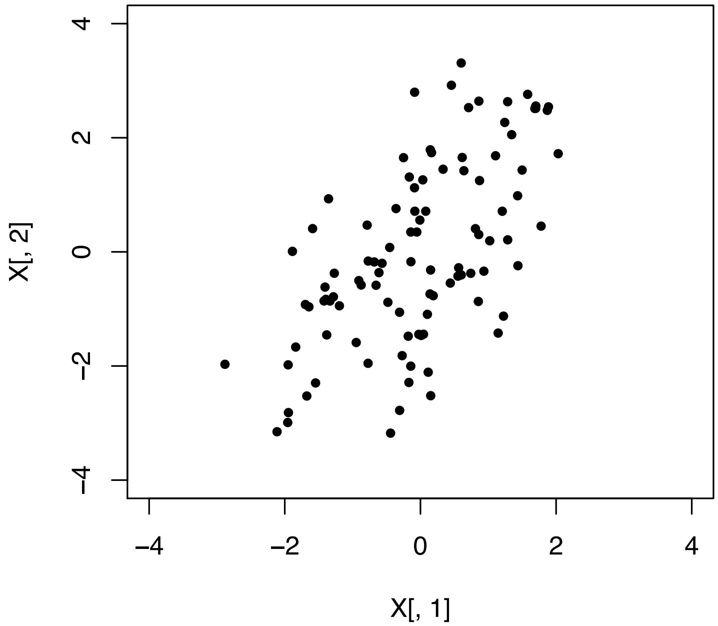
no

the intercepts change



Thinking about the decision function

Let's look at the $(x_i - \mu_k)^T \Sigma^{-1} (x_i - \mu_k)$ term that appears in the LDA decison rule. How should we think about this?



Mahalanobis distance

Mahalanobis distance measures the distance from a center in terms of the variance of the distribution.

For a Gaussian, it captures how far out in the tail of the distribution a point is. If $(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$ is large then $p(x_i)$ is small.

This is used for LDA, QDA, and Multivariate Gaussian distributions.

It can also be used for outlier detection and for clustering!

Suppose I've seen a lot of data, and now a new point comes along. How can I tell whether it's a "rare" or outlier point?

Mahalanobis distance, PCA, and LDA

Note that LDA equivalently minimizes over $k = 1, \dots K$,

$$\frac{1}{2}(x - \widehat{\mu}_k)^T \widehat{\Sigma}^{-1}(x - \widehat{\mu}_k) - \log \widehat{\pi}_k$$

It helps to factorize $\widehat{\Sigma}$ (i.e., compute its eigendecomposition):

$$\widehat{\Sigma} = UDU^T$$

where $U \in \mathbb{R}^{p \times p}$ has orthonormal columns (and rows), and $D = \mathrm{diag}(d_1, \dots d_p)$ with $d_k \geq 0$ for each k. Then we have $\widehat{\Sigma}^{-1} = UD^{-1}U^T$, and

$$(x - \widehat{\mu}_k)^T \widehat{\Sigma}^{-1}(x - \widehat{\mu}_k) = (x-\hat{\mu}_k)^T U D^{-1/2} D^{-1/2} U^T (x-\hat{\mu}_k)$$

$$= (\tilde{x} - \tilde{\hat{\mu}}_k)^T (\tilde{x} - \tilde{\hat{\mu}}_k) = \|\tilde{x} - \tilde{\hat{\mu}}_k\|_2^2$$

This is just the squared distance between $\tilde{x}$ and $\tilde{\mu}_k$!

$\tilde{x} = D^{-1/2} U^T x \qquad \tilde{\hat{\mu}}_k = D^{-1/2} U^T \hat{\mu}_k.$

Sphering

What is this transformation doing? Think about applying it to the observations:

$$\tilde{x}_i = D^{-1/2} U^T x_i, \quad i = 1, \dots n$$

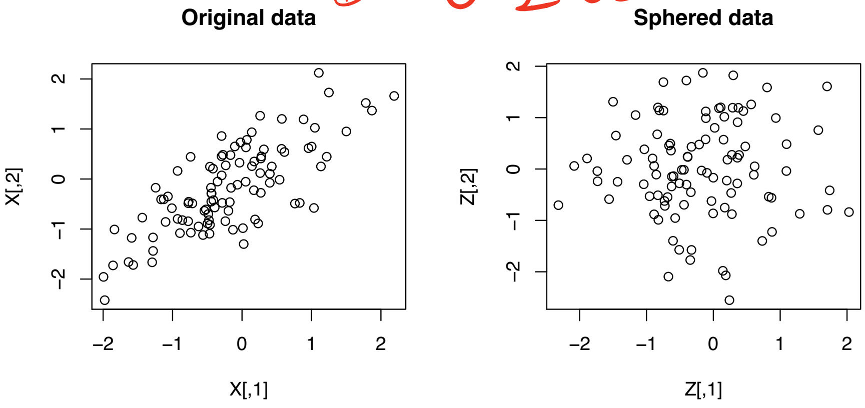
This is basically sphering the data points, because if we think of $x \in \mathbb{R}^p$ were a random variable with covariance matrix $\widehat{\Sigma}$, then

$$\mathrm{Cov}(D^{-1/2} U^T x) = \mathbb{E}\, D^{-1/2} U^T (x-\mu)(x-\mu)^T U D^{-1/2}$$

$$= D^{-1/2} U^T \hat{\Sigma} U D^{-1/2} = D^{-1/2} U^T U D U^T U D^{-1/2}$$

$$= D^{-1/2} D D^{-1/2}$$

$$= I.$$



LDA transformed

Hence the LDA procedure can be described as:

1. Compute the sample estimates $\widehat{\pi}_k, \widehat{\mu}_k, \widehat{\Sigma}$

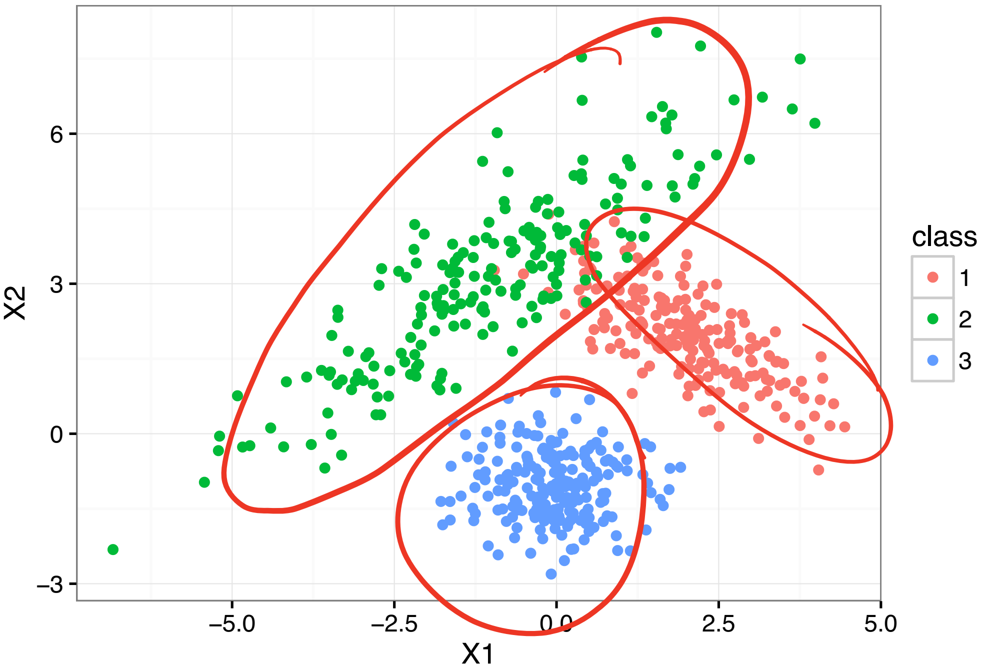
2. Factor $\widehat{\Sigma}$, as in $\widehat{\Sigma} = UDU^T$

3. Transform the class centroids $\tilde{\mu}_k = D^{-1/2} U^T \widehat{\mu}_k$

4. Given any point $x \in \mathbb{R}^p$, transform to $\tilde{x} = D^{-1/2} U^T x \in \mathbb{R}^p$, and then classify according to the nearest centroid in the transformed space, adjusting for class proportions—this is the class k for which $\frac{1}{2}\|\tilde{x} - \tilde{\mu}_k\|_2^2 - \log \widehat{\pi}_k$ is smallest

Variations on LDA: unequal Σ

The LDA model assumes that our covariance is the same for all groups. What if it's not?



$$\Sigma_1 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}, \; \Sigma_2 = \begin{pmatrix} 3 & 2.5 \\ 2.5 & 3 \end{pmatrix}, \; \Sigma_3 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

All the same math goes through, giving Quadratic Discriminant Functions (and QDA)

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\Sigma_k^{-1}x + \mu_k\Sigma_k^{-1}x - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k\end{aligned}$$

The Σ matrices are now different in each function are now different, and the boundaries $\delta_k(x) > \delta_{k'}$ are no longer linear. Instead, they are quadratic

This allows the boundaries to curve around groups to account for different patterns of spread. It comes at a cost though:

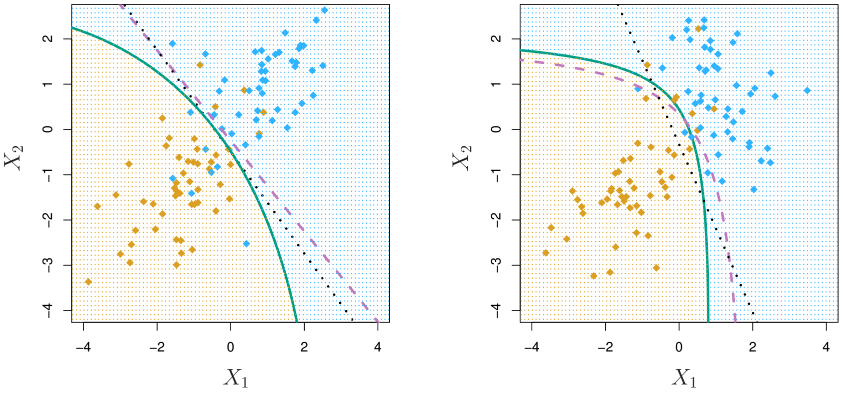
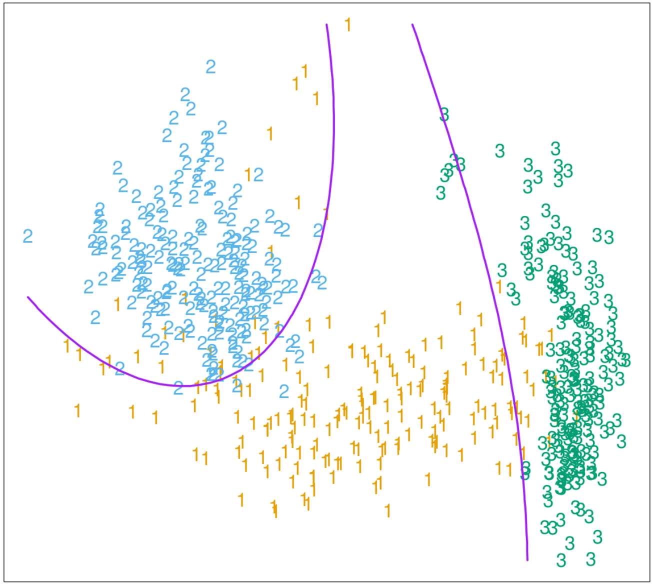


Figure 4.9 from ISL. Dashed purple curve is the Bayes classifier decision boundary. Solid green curve is QDA, dotted black line is LDA. Left: True boundary is linear. Right: True boundary is quadratic

Variations on LDA: unequal Σ

Quadratic discriminant analysis (QDA):



(ESL 4.6)

Variations on LDA: high dimensions

LDA really becomes instructive when we consider its performance over a range of dimensions.

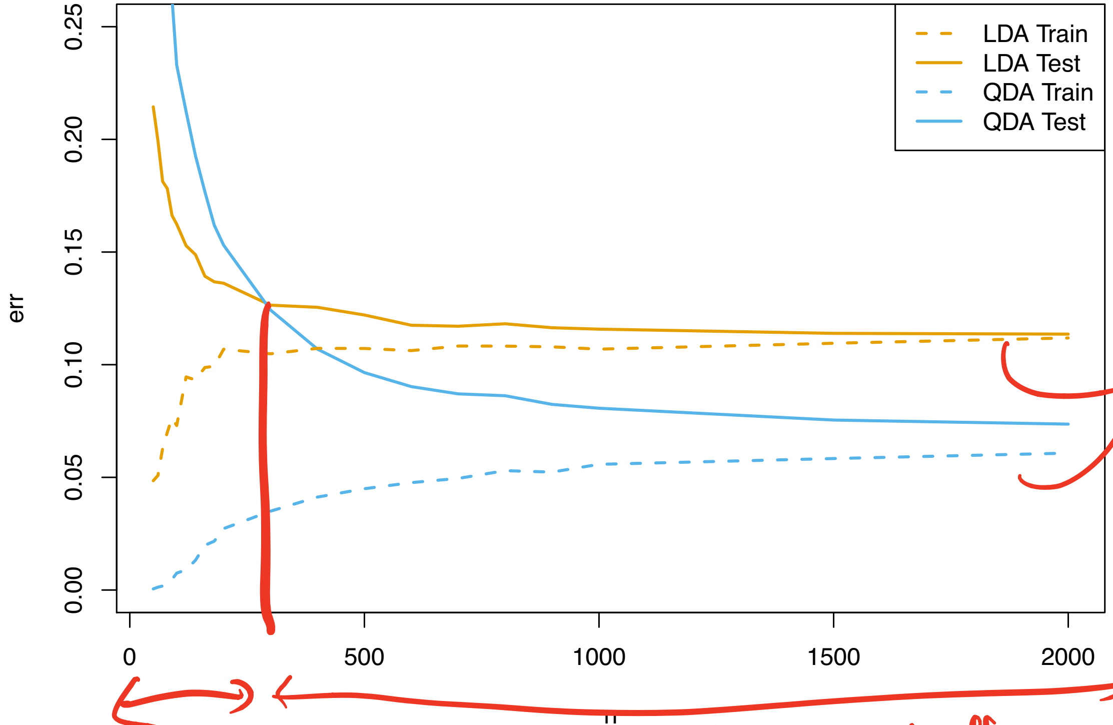
Fitting QDA requires estimating
- k means - kp
- k covariance matrices - $\frac{kp(p+1)}{2}$

Fitting LDA requires estimating
- kp
- $p(p+1)/2$

As the number of parameters increases, the variance of our estimator increases (but the bias hopefully decreases).

Suppose that we have two groups, drawn respectively from $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$. If we choose to use LDA instead of QDA, we are choosing a more biased model to reduce variance.

Variations on LDA: high dimensions



train error of QDA < LDA

LDA better

QDA better

Variations on LDA: high dimensions

Suppose the dimension gets even higher? → Approximate Σ by a diagonal matrix (only est. variances)

→ reduce parameters from $O(p^2) \rightarrow O(p)$.

What if I can't even estimate the group means well? Try regularizing with LASSO.