Linear Methods for Classification

-Statistical Machine Learning-

Lecturer: Darren Homrighausen, PhD

An Overview of Classification

Some examples:

- A person arrives at an emergency room with a set of symptoms that could be 1 of 3 possible conditions. Which one is it?
- A online banking service must be able to determine whether each transaction is fraudulent or not, using a customer's location, past transaction history, etc.
- Given a set of individuals sequenced DNA, can we determine whether various mutations are associated with different phenotypes?

All of these problems are not regression problems. They are classification problems.

The Set-up

It begins just like regression: suppose we have observations

$$\mathcal{D} = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

Again, we want to estimate a function that maps X into Y that helps us predict as yet observed data.

(This function is known as a classifier)

The same constraints apply:

- We want a classifier that predicts test data, not just the training data.
- Often, this comes with the introduction of some bias to get lower variance and better predictions.

How do we measure quality?

In regression, we have $Y_i \in \mathbb{R}$ and use squared error loss

Instead, let $Y \in \mathcal{G} = \{1, \ldots, G\}$

(This is arbitrary, sometimes other numbers, such as $\{-1, 1\}$ will be used)

We again make predictions $\hat{Y}$ based on $\mathcal{D}$

Our loss function is now a $G \times G$ matrix L with

- zeros on the diagonals
- $L_{g,g'} > 0$ on the off diagonal $(g \neq g')$

($L_{g,g'} = \mathbf{1}_{Y \neq g(X)}$ is zero-one loss)

We will use the loss function notation

$$\ell_g(Z) = L_{g,Y}$$

which is implicitly a function of the feature X

How do we measure quality?

Again, we appeal to risk

$$R(g) = \mathbb{E}_Z \ell_g(Z)$$

If we use the law of total probability, this can be written

$$R(g) = \mathbb{E}_X \sum_{y=1}^{G} \ell_g(Z = (y, X))\mathbb{P}(Y = y|X)$$

This can be minimized point wise over X, to produce

$$g_*(X) = \operatorname*{argmin}_{g \in \mathcal{G}} \sum_{y=1}^{G} \ell_g(Z = (y, X))\mathbb{P}(Y = y|X)$$

(This is the Bayes' classifier. Also, $R(g_*)$ is the Bayes' limit)

Best classifier

If we make specific choices for ℓ, we can find g_* exactly

As Y takes only a few values, zero-one prediction risk is natural

$$\ell_g(Z) = \mathbf{1}_{Y \neq g(X)}(Z) \Rightarrow R(g) = \mathbb{E}[\ell_g(Z)] = \mathbb{P}(g(X) \neq Y),$$

(This means we want to label or classify a new observation (X, Y) such that $g(X) = Y$ as often as possible)

Under this loss, we have

$$g_*(X) = \operatorname*{argmin}_{g \in \mathcal{G}} [1 - \mathbb{P}(Y = g|X)] = \arg\max_{g \in \mathcal{G}} \mathbb{P}(Y = g|X)$$

Best classifier

Suppose we encode a two-class supervisor as $Y \in \{0, 1\}$

Let's continue to use squared error loss: $\ell_f(Z) = (Y - f(X))^2$

Then, the Bayes' rule is

$$f_*(X) = \mathbb{E}[Y|X] = \mathbb{P}(Y = 1|X)$$

(using f as it references squared error loss)

Hence, we achieve the same Bayes' rule/limit as 0-1 loss with squared error loss by discretizing the regression function:

$$g_*(X) = \mathbf{1}(f_*(X) > 1/2)$$

Classification is easier than regression

Let $\hat{f}$ be any estimate of f_*

Let $\hat{g}(X) = \mathbf{1}(\hat{f}(X) > 1/2)$

It can be shown that

$$\begin{aligned}
&\mathbb{P}(Y \neq \hat{g}(X)|X) - \mathbb{P}(Y \neq g_*(X)|X) \\
&= (2f_*(X) - 1)(\mathbf{1}(g_*(X) = 1) - \mathbf{1}(\hat{g}(X) = 1)) \\
&= |2f_*(X) - 1|\mathbf{1}(g_*(X) \neq \hat{g}(X)) \\
&= 2\left|f_*(X) - \frac{1}{2}\right|\mathbf{1}(g_*(X) \neq \hat{g}(X))
\end{aligned}$$

[Can you show this?]

Classification is easier than regression

Now

$$g_*(X) \neq \hat{g}(X) \Rightarrow |\hat{f}(X) - f_*(X)| \geq |\hat{f}(X) - 1/2|$$

Therefore

$$\begin{aligned}
&\mathbb{P}(Y \neq \hat{g}(X)) - \mathbb{P}(Y \neq g_*(X)) = \\
&= \int (\mathbb{P}(Y \neq \hat{g}(X)|X) - \mathbb{P}(Y \neq g_*(X)|X)) d\mathbb{P}_X \\
&= \int 2 \left| \hat{f}(X) - \frac{1}{2} \right| \mathbf{1}(g_*(X) \neq \hat{g}(X)) d\mathbb{P}_X \\
&\leq 2 \int |\hat{f}(X) - f_*(X)| \mathbf{1}(g_*(X) \neq \hat{g}(X)) d\mathbb{P}_X \\
&\leq 2 \int |\hat{f}(X) - f_*(X)| d\mathbb{P}_X
\end{aligned}$$

Classification is easier than regression

Now

$$g_*(X) \neq \hat{g}(X) \Rightarrow |\hat{f}(X) - f_*(X)| \geq |\hat{f}(X) - 1/2|$$

Therefore

$$\begin{aligned}
&\mathbb{P}(Y \neq \hat{g}(X)) - \mathbb{P}(Y \neq g_*(X)) = \\
&= \int (\mathbb{P}(Y \neq \hat{g}(X)|X) - \mathbb{P}(Y \neq g_*(X)|X)) d\mathbb{P}_X \\
&= \int 2 \left| \hat{f}(X) - \frac{1}{2} \right| \mathbf{1}(g_*(X) \neq \hat{g}(X)) d\mathbb{P}_X \\
&\leq 2 \int |\hat{f}(X) - f_*(X)| \mathbf{1}(g_*(X) \neq \hat{g}(X)) d\mathbb{P}_X \\
&\leq 2 \int |\hat{f}(X) - f_*(X)| d\mathbb{P}_X
\end{aligned}$$

(If $\hat{f}$ gets close to f_* on average, we do good classifications. The converse is not true)

Bayes' rule and class densities

Using Bayes' theorem

$$\begin{aligned} f_*(X) &= \mathbb{P}(Y = 1|X) \\ &= \frac{p(X|Y = 1)\mathbb{P}(Y = 1)}{\sum_{g \in \{0,1\}} p(X|Y = g)\mathbb{P}(Y = g)} \\ &= \frac{f_1(X)\pi}{f_1(X)\pi + f_0(X)(1 - \pi)} \end{aligned}$$

We call $f_g(X)$ the class densities

The Bayes' rule can be rewritten

$$g_*(X) = \begin{cases} 1 & \text{if } \frac{f_1(X)}{f_0(X)} > \frac{1-\pi}{\pi} \\ 0 & \text{otherwise} \end{cases}$$

How to find a classifier

All of these prior expressions for g_* give rise to classifiers

- Empirical risk minimization: Choose a set of classifiers Γ and find $\hat{g} \in \Gamma$ that minimizes some estimate of $R(g)$

  (This can be quite challenging as, unlike in regression, the training error is nonconvex)
- Regression: Find an estimate $\hat{f}$ and plug it in to the Bayes' rule
- Density estimation: Estimate $\hat{\pi}$ and f_g from $\mathcal{D}$ where $Y = g$ and

Linear classifiers

Linear classifier

As our classifier $\hat{g}$ takes a discrete number of values, it is equivalent to partitioning the feature space into regions

The boundaries between these regions are known as decision boundaries

These decision boundaries are sets of points at which $\hat{g}$ is indifferent between two (or more) classes

A linear classifier is a $\hat{g}$ that produces linear decision boundaries

Linear classifier: Example

Suppose $\mathcal{G} = \{-1, 1\}$ and we form the GLM logistic regression

The posterior probabilities are

$$\mathbb{P}(Y = 1|X) = \frac{\exp\{\beta_0 + \beta^\top X\}}{1 + \exp\{\beta_0 + \beta^\top X\}}$$
$$\mathbb{P}(Y = -1|X) = \frac{1}{1 + \exp\{\beta_0 + \beta^\top X\}}$$

The logit (i.e.: log odds) transformation forms a linear decision boundary

$$\log\left(\frac{\mathbb{P}(Y = 1|X)}{\mathbb{P}(Y = -1|X)}\right) = \beta_0 + \beta^\top X$$

The decision boundary is the hyperplane $\{X : \beta_0 + \beta^\top X = 0\}$

(Log-odds below 0, classify as -1, above 0 classify as a 1)

Linear classifier: Extensions

The term "linear classifier" can be used to describe a classifier that has linear decision boundaries in a higher dimensional space, but which as a nonlinear decision boundary in the original feature space

For instance, if I include as features:

$$x_1^2, \ldots, x_p^2, x_1 x_2, \ldots, x_1 x_p, \ldots$$

and thereby add $p(p+1)/2$ additional features, a linear classifier in this enhanced space will be nonlinear (and in fact quadratic) in the original features

This is a parametric kernel method

Bayes' rule-ian approach

The decision theory for classification indicates we need to know the posterior probabilities: $\mathbb{P}(Y = g|X)$ for doing optimal classification

Suppose that

- $p_g(X) = \mathbb{P}(X|Y = g)$ is the likelihood of the features given the class labels
- $\pi_g = \mathbb{P}(Y = g)$ is the prior

Then

$$\mathbb{P}(Y = g|X) = \frac{p_g(X)\pi_g}{\sum_{g \in \mathcal{G}} p_g(X)\pi_g} \propto p_g(X)\pi_g$$

Conclusion: Having the class densities almost gives us the Bayes' rule as the training proportions can usually be used to estimate π_g

(Though, sometimes estimating π_g can be nontrivial/impossible)

Bayes' rule-ian approach: Summary

There are many techniques based on this idea

- Linear/quadratic discriminant analysis
  (Estimates p_g assuming multivariate Gaussianity)
- General nonparametric density estimators
- Naive Bayes (Factors p_g assuming conditional independence)

Discriminant analysis

Suppose that

$$p_g(X) \propto |\Sigma_g|^{-1/2} e^{-(X-\mu_g)^\top \Sigma_g^{-1}(X-\mu_g)/2}$$

Let's assume that $\Sigma_g \equiv \Sigma$.

Then the log-odds between two classes g, g' is:

$$\begin{aligned}\log\left(\frac{\mathbb{P}(Y=g|X)}{\mathbb{P}(Y=g'|X)}\right) &= \log\frac{p_g(X)}{p_{g'}(X)} + \log\frac{\pi_g}{\pi_{g'}} \\ &= \log\frac{\pi_g}{\pi_{g'}} - (\mu_g+\mu_{g'})^\top\Sigma^{-1}(\mu_g-\mu_{g'})/2 \\ &\quad + X^\top\Sigma^{-1}(\mu_g-\mu_{g'})\end{aligned}$$

This is linear in X, and hence has a linear decision boundary

Types of discriminant analysis

The linear discriminant function is (proportional to) the log posterior:

$$\delta_g(X) = \log \pi_g + X^\top \Sigma^{-1} \mu_g - \mu_g^\top \Sigma^{-1} \mu_g / 2$$

and we assign $g(X) = \arg\max_g \delta_g(X)$

Linear/regularized discriminant analysis

Now, we must estimate μ_g and Σ. If we...

- use the intuitive estimators $\hat{\mu}_g = \overline{X}_g$ and

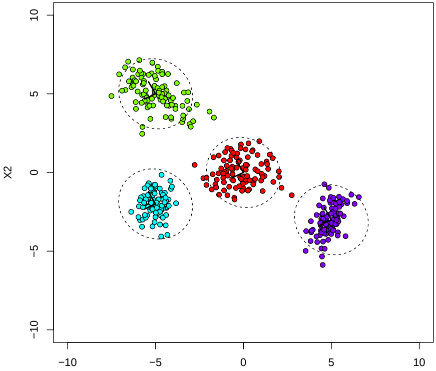
  $$\hat{\Sigma} = \frac{1}{n - G} \sum_{g \in \mathcal{G}} \sum_{i \in g} (X_i - \hat{\mu}_g)(X_i - \hat{\mu}_g)^\top$$

  then we have produced linear discriminant analysis (LDA)
- regularize these ‘plug-in’ estimates, we can form regularized discriminant analysis (Friedman (1989)). This could be (for $\lambda \in [0, 1]$):

  $$\hat{\Sigma}_\lambda = \lambda \hat{\Sigma} + (1 - \lambda)\hat{\sigma}^2 I$$

LDA intuition

How would you classify a point with this data?



We can just classify an observation to the closest mean ($\overline{X}_g$)

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

LDA intuition

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

$$\begin{aligned}\tilde{g}(X) &= \underset{g}{\operatorname{argmin}} ||X - \overline{X}_g||_2^2 \\ &= \underset{g}{\operatorname{argmin}} X^\top X - 2X^\top \overline{X}_g + \overline{X}_g^\top \overline{X}_g \\ &= \underset{g}{\operatorname{argmin}} -X^\top \overline{X}_g + \frac{1}{2}\overline{X}_g^\top \overline{X}_g\end{aligned}$$

compare this to:

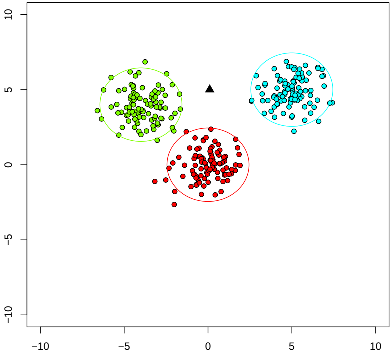
$$\hat{g} = \underset{g}{\arg\max} \underbrace{X^\top \hat{\Sigma}_\lambda^{-1} \overline{X}_g - \frac{1}{2}\overline{X}_g^\top \hat{\Sigma}_\lambda^{-1} \overline{X}_g}_{\textit{likelihood}} + \underbrace{\log(\hat{\pi}_g)}_{\textit{prior}}$$

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

(Note: this generalization of Euclidean distance is called Mahalanobis distance)

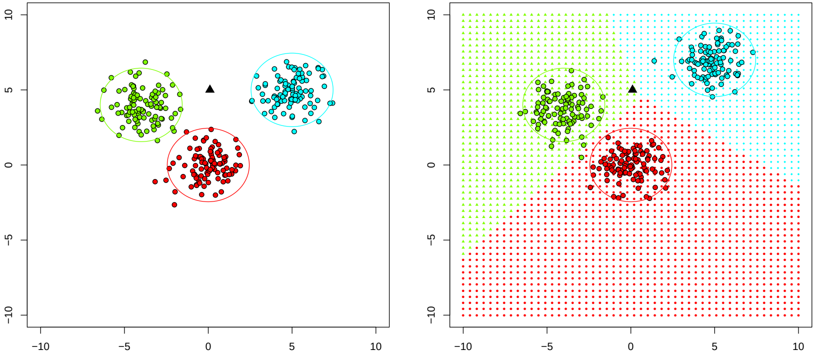
Intuition

What if the data looked like this?



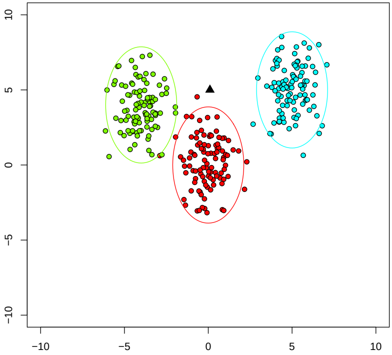
Intuition

What if the data looked like this?



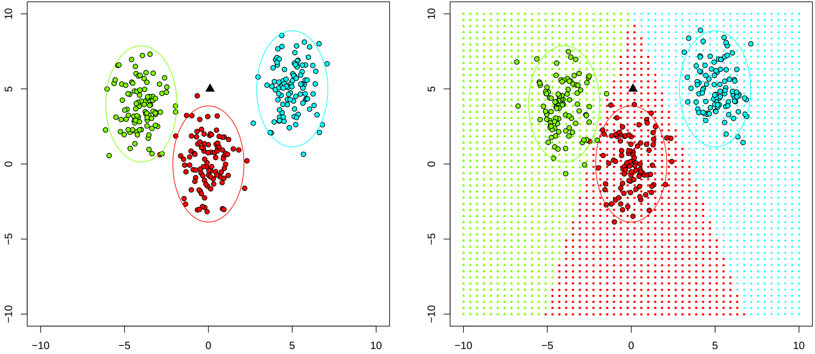
Intuition

Or this?



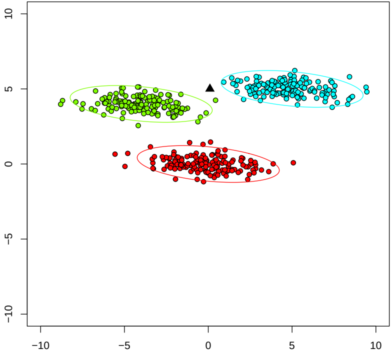
Intuition

Or this?



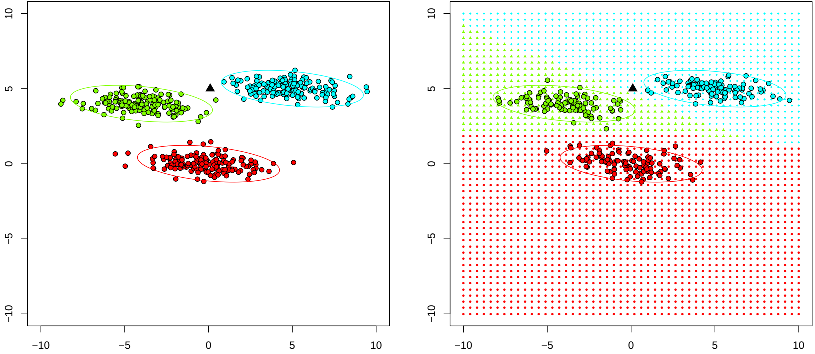
Intuition

How about this?



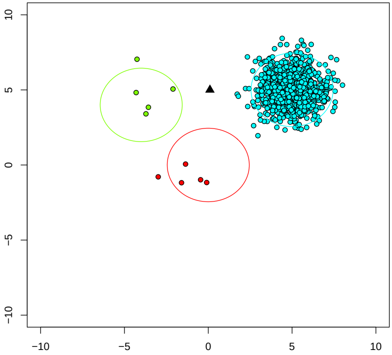
Intuition

How about this?



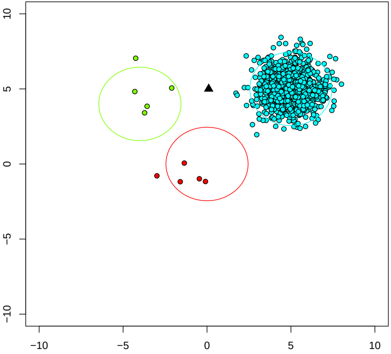
Intuition

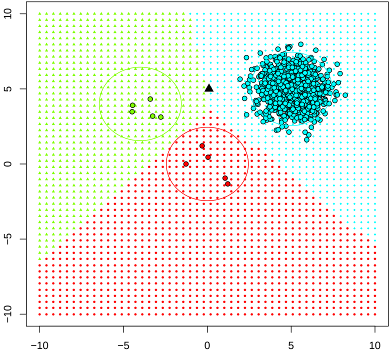
What about now?



Intuition

What about now?





Performance of LDA

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

- The sample size n

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

  (That is: $X|Y = g$ is a Gaussian with mean μ_g and variance Σ)

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

LDA: under correct assumptions

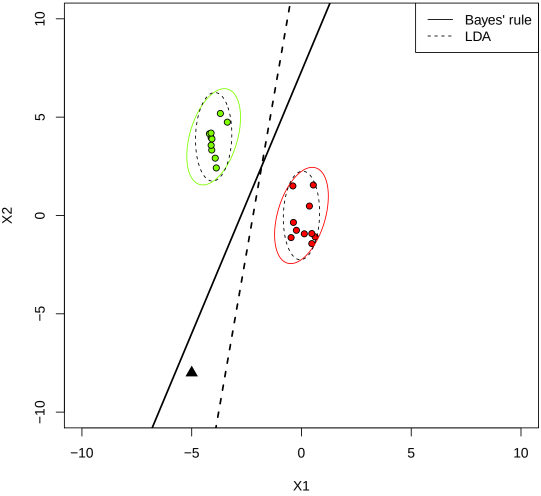


FIGURE: For $n_g = 10$

LDA: under correct assumptions

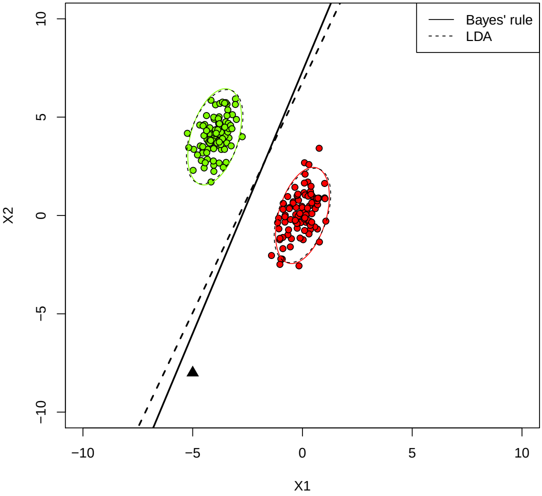


Figure: For $n_g = 100$

LDA: under correct assumptions

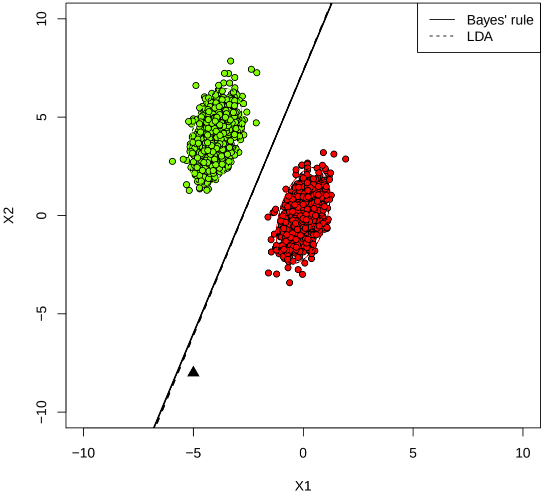


FIGURE: For $n_g = 1000$

LDA: mildly incorrect assumptions

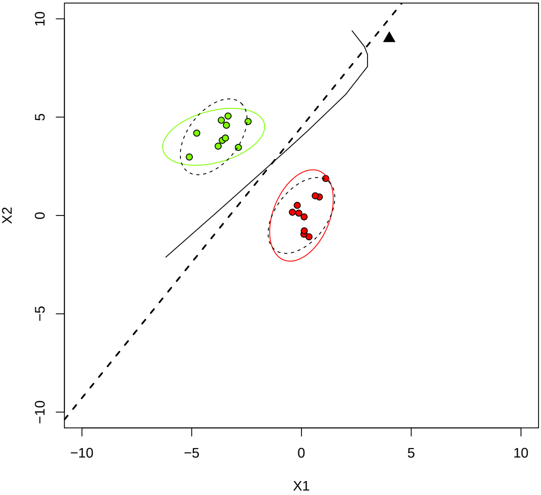


Figure: For $n_g = 10$

LDA: mildly incorrect assumptions

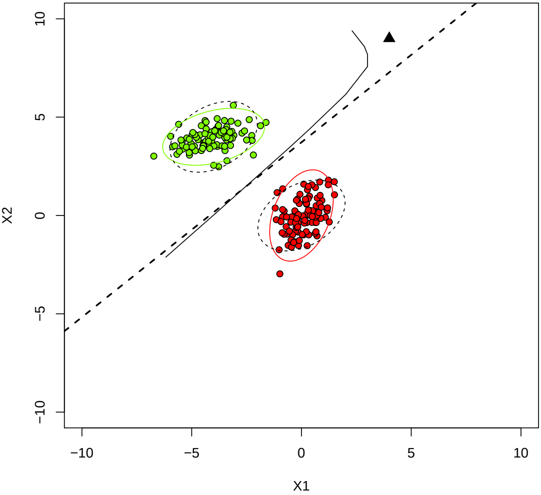


Figure: For $n_g = 100$

LDA: mildly incorrect assumptions

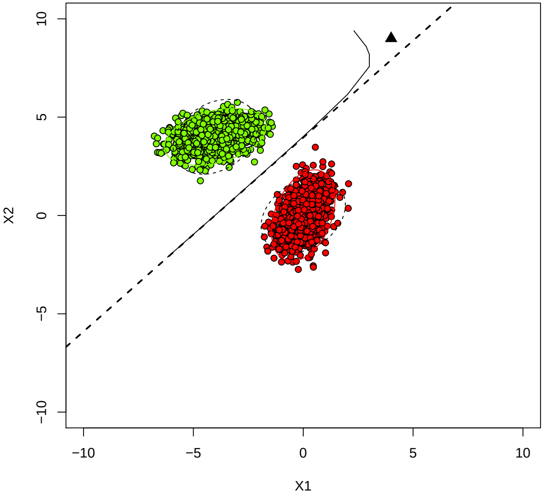


Figure: For $n_g = 1000$

LDA: very incorrect assumptions

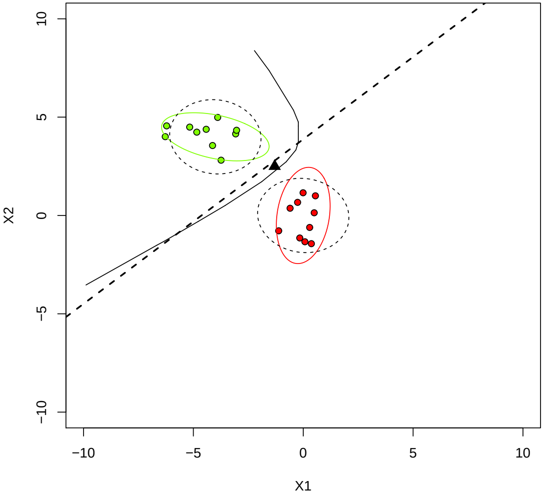


Figure: For $n_g = 10$

LDA: very incorrect assumptions

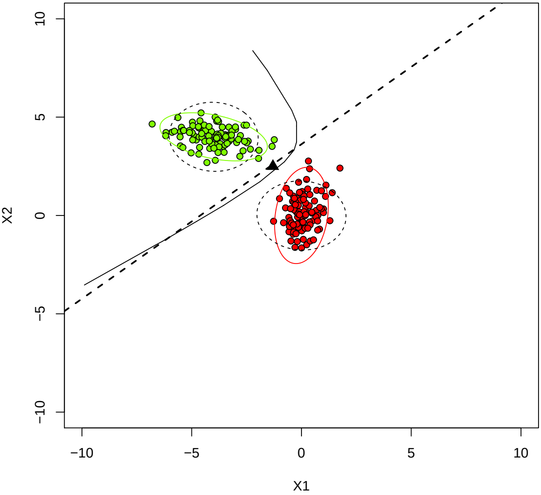


Figure: For $n_g = 100$

LDA: very incorrect assumptions

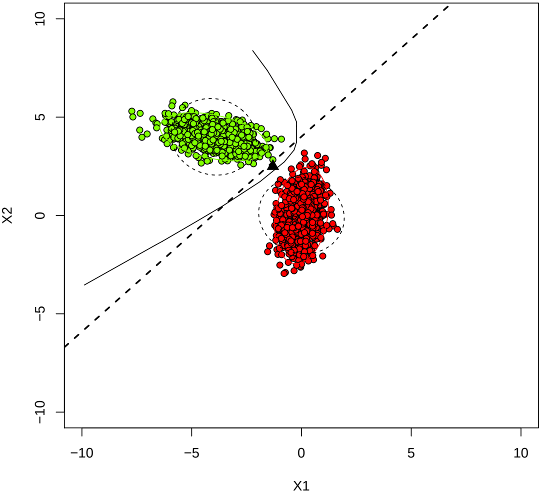


Figure: For $n_g = 1000$

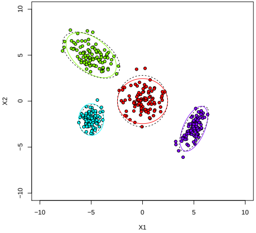
The LDA variance assumption

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

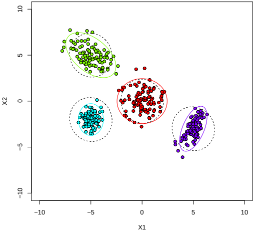
The assumption provides two benefits:

- Allows for estimation when n isn't large compared with $Gp(p+1)/2$
- Lowers the variance of the procedure (but produces bias)

(This can be seen by the estimation of fewer parameters)



Different $\hat{\Sigma}_g$



All same $\hat{\Sigma}$

The LDA variance assumption

However, when n is large compared with $Gp(p+1)/2$

(Say, $\min n_g \geq 40p(p+1)/2$)

Then the induced bias can outweigh the variance

(This is hard to determine. Usually compare the prediction error on test set)

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

- mean μ_g
- variance Σ_g

This makes the decision boundary quadratic

(Instead of linear)

Quadratic Discriminant Analysis

Quadratic discriminant analysis

If we drop the assumption regarding equal covariances, we get:

$$\delta_g(X) = \log \pi_g + X^\top \Sigma_g^{-1} \mu_g - \mu_g^\top \Sigma_g^{-1} \mu_g / 2 - \log |\Sigma_g| / 2$$

(Σ_g can be estimated by the sample covariance of the observations in group g)

This produces quadratic discriminant analysis (QDA)

In my experience, QDA works well if n is large relative to p

(However, it isn't often computable in practice; too many parameters)

We can augment regularized discriminant analysis to shrink each $\hat{\Sigma}_g$ to $\hat{\Sigma}$ or even to $\hat{\Sigma}_\lambda$

$$\hat{\Sigma}_{g,(\gamma,\lambda)} = \gamma \hat{\Sigma}_g + (1 - \gamma) \hat{\Sigma}_\lambda$$

(To the best of my knowledge, little is formally known about this procedure. See Guo et al. (2006) for an empirical comparison)

QDA: More flexibility than needed

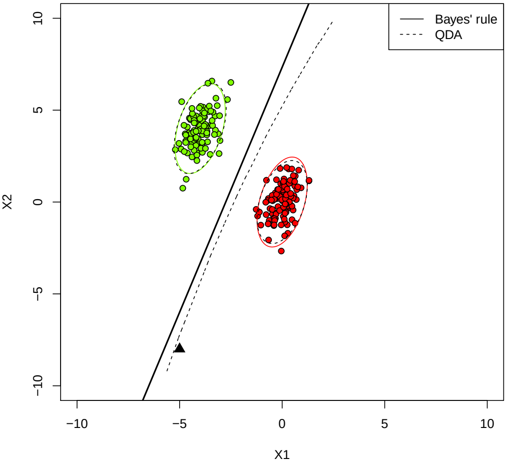


Figure: For $n_g = 100$. Note linear Bayes' rule, nonlinear QDA decision boundary

QDA: More flexibility than needed

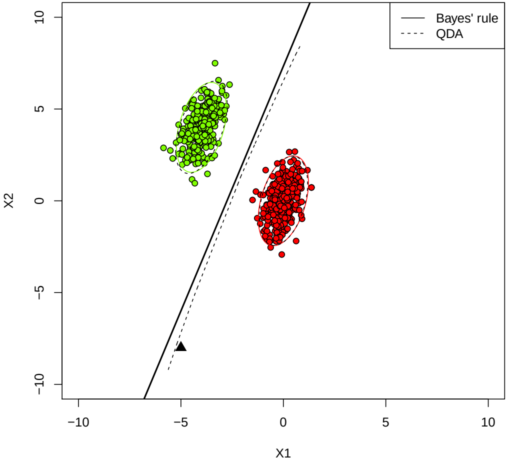


Figure: For $n_g = 300$. Note linear Bayes' rule, nonlinear QDA decision boundary

QDA: More flexibility than needed

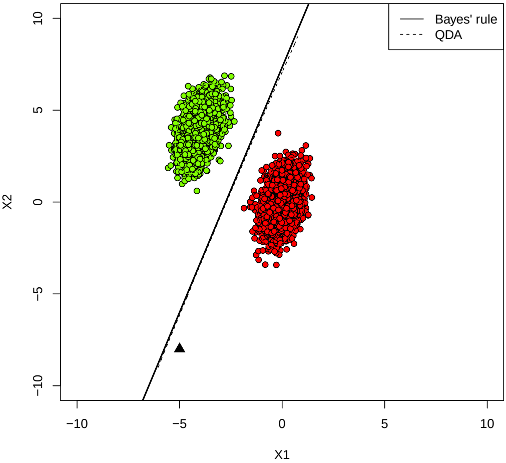


Figure: For $n_g = 2000$. Note linear Bayes' rule. The nonlinear QDA decision boundary has converged to Bayes' rule

QDA: Different Σ_g assumption needed

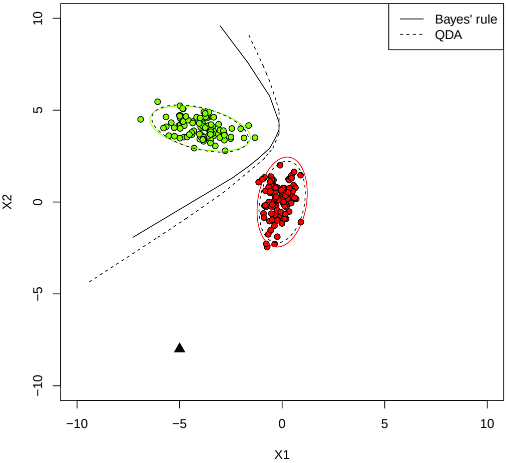


Figure: For $n_g = 100$. Note nonlinear Bayes' rule, nonlinear QDA decision boundary

QDA: Different Σ_g assumption needed

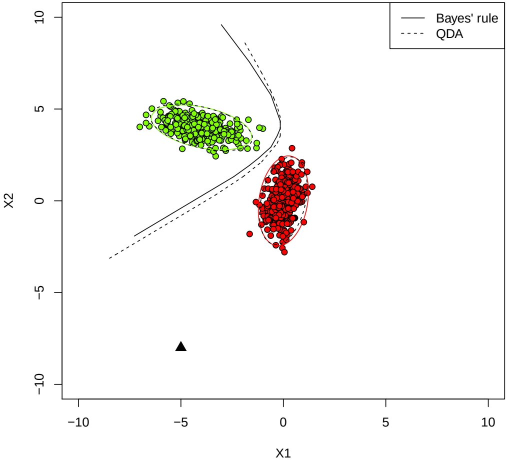


Figure: For $n_g = 300$. Note nonlinear Bayes' rule, nonlinear QDA decision boundary

QDA: Different Σ_g assumption needed

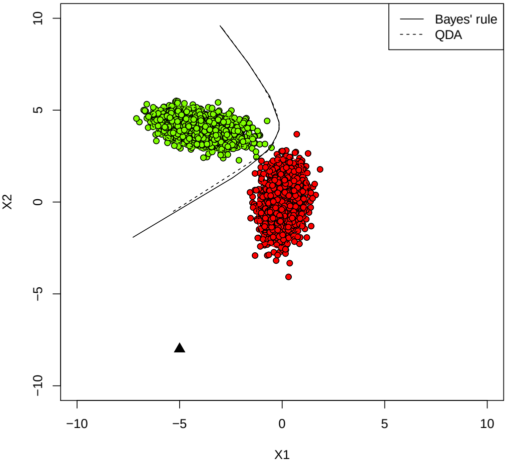


Figure: For $n_g = 2000$. Note nonlinear Bayes' rule, nonlinear QDA decision boundary

LDA vs. QDA: under correct assumptions

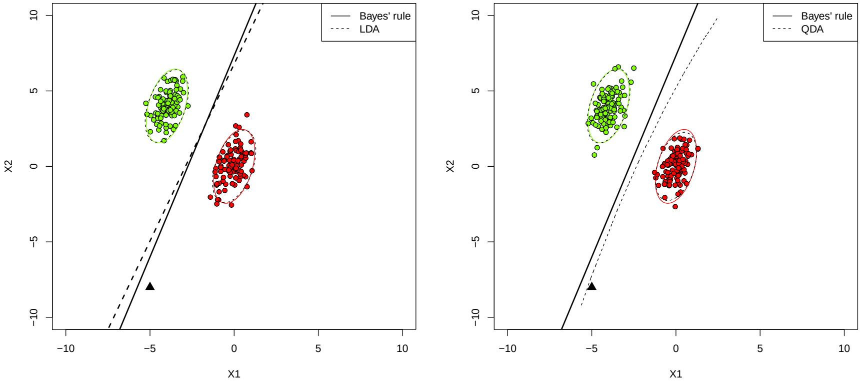


Figure: For $n_g = 100$

LDA vs. QDA: very incorrect assumptions

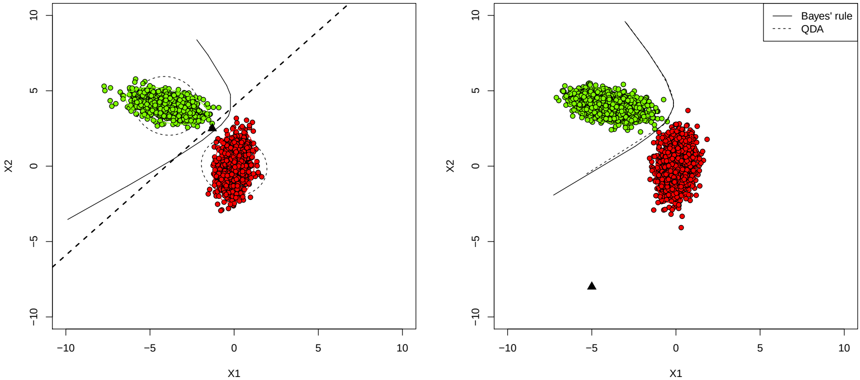


FIGURE: LDA $n_g = 1000$, QDA $n_g = 2000$

LDA in R

We can do this readily in R

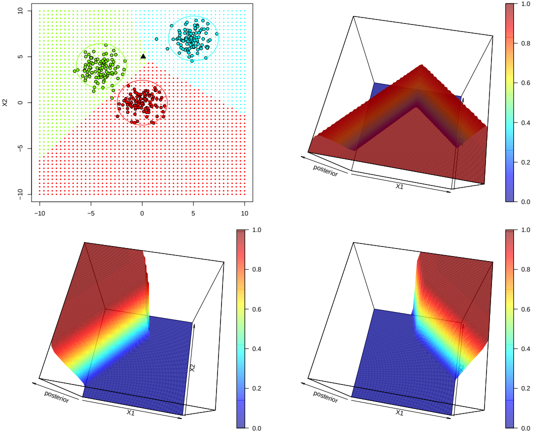
```
library(MASS)
lda.fit  = lda(Y~.,data=X)

> 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
```

Reduced rank LDA

Reduced rank LDA

Part of the popularity of LDA is that it provides dimension reduction as well

The G class centroids μ_g must all lie in an affine subspace of dimension $G - 1$ (presuming $G < p$)

(Let $\mathcal{H}_{G-1}$ be this subspace)

If G is much less than p, this will be a substantial drop in dimension

Reduced rank LDA

In practice, we can compute LDA from spectral information:

$$\begin{aligned}\delta_g(X) &= \log \pi_g + X^\top \Sigma^{-1} \mu_g - \mu_g^\top \Sigma^{-1} \mu_g / 2 \\ &\propto \log \pi_g + (X - \mu_g)^\top \Sigma^{-1} (X - \mu_g)/2\end{aligned}$$

So,

1. Spectrum: Form $\hat{\Sigma}_\lambda = UDU^\top$
2. Sphere: Rewrite your data as $\tilde{X} \leftarrow D^{-1/2} U^\top X$
3. Assign: Classify to the closest mean in transformed space

   (Penalizing by estimate of prior probability)

Reduced rank LDA

We can ignore any information orthogonal to $\mathcal{H}_{G-1}$, as it contributes to each class equally (in the sphered space)

So, project $\tilde{X}$ onto $\mathcal{H}_{G-1}$ and make distance computations there

When $G = 2, 3$, this means we can plot the projection onto $\mathcal{H}_{G-1}$ with no loss of information about the LDA solution

If $G > 3$, then we may wish to project onto a reduced space $\mathcal{H}_L \subset \mathcal{H}_{G-1}$

We'd like $\mathcal{H}_L$ to maintain the most amount of information possible for assigning to classes

Reduced rank LDA

This can be done via the following procedure

1. Centroids: Compute $G \times p$ matrix M of class centroids
2. Covariance: Form $\hat{\Sigma}$ as the common covariance matrix
3. Sphere: $\tilde{M} = M\hat{\Sigma}^{-1/2}$
4. Between Covariance: Find covariance matrix for $\tilde{M}$, call it B
5. Spectrum Compute $B = VSV^\top$

(These later two steps are getting the PCA for the empirical means)

Now, $\text{span}(V_L) = \mathcal{H}_L$

Also, the coordinates of the data in this space are $Z_k = v_k^\top \hat{\Sigma}^{-1/2} X$

These derived variables are commonly called canonical coordinates

Reduced rank LDA: Summary

- Gaussian likelihoods with identical covariances leads to linear decision boundaries (LDA)
- We can actually do all relevant computations/graphics on the reduced space $\mathcal{H}_{G-1}$
- If this isn't small enough, we can do 'optimal' dimension reduction to $\mathcal{H}_L$

As an aside, this procedure is identical to Fisher's discriminant analysis

Logistic regression

Logistic regression for two classes simplifies to a likelihood:

(Using $\pi_i(\beta) = \mathbb{P}(Y = 1|X = X_i, \beta)$)

$$\begin{aligned}\ell(\beta) &= \sum_{i=1}^{n} (Y_i \log(\pi_i(\beta)) + (1 - Y_i) \log(1 - \pi_i(\beta))) \\ &= \sum_{i=1}^{n} \left(Y_i \log(e^{\beta^\top X_i}/(1 + e^{\beta^\top X_i})) - (1 - Y_i) \log(1 + e^{\beta^\top X_i}) \right) \\ &= \sum_{i=1}^{n} \left(Y_i \beta^\top X_i - \log(1 + e^{\beta^\top X_i}) \right)\end{aligned}$$

This gets optimized via Newton-Raphson updates and iteratively reweighted least squares

Sparse logistic regression

This procedure suffers from all the same problems as least squares

We can use penalized likelihood techniques in the same way as we did before

This means maximizing (over β_0, β):

$$\sum_{i=1}^{n} \left(Y_i(\beta_0 + \beta^\top X_i) - \log(1 + e^{\beta_0 + \beta^\top X_i}) \right) - \lambda(\alpha||\beta||_1 + (1-\alpha)||\beta||_2^2)$$

(Don't penalize the intercept and do standardize the features)

This is the logistic elastic net

Sparse logistic regression: Software

Using the R package glmnet finds the minimum CV solution over a grid of λ values

Unfortunately, the computations are more difficult for path algorithms (such as the lars package) due to the coefficient profiles being only piecewise smooth

glmpath is an R package that does quadratic approximations to the profiles, while still computing the exact points at which the active set changes

(Park, Hastie (2007). It is necessary to set a 'step' size argument for the approximation.)

Logistic versus LDA

The log posterior odds via the Gaussian likelihood (LDA) for class g versus G are

$$\begin{aligned}\log\frac{\mathbb{P}(Y=g|X)}{\mathbb{P}(Y=G|X)} &= \log\frac{\pi_g}{\pi_G}-(\mu_g+\mu_G)^\top\Sigma^{-1}(\mu_g-\mu_G)/2\\ &\quad + X^\top\Sigma^{-1}(\mu_g-\mu_G)\\ &= \alpha_{g,0}+\alpha_g{}^\top X\end{aligned}$$

Likewise, multi class logistic follows (for $g=1,\ldots,G-1$):

$$\log\frac{\mathbb{P}(Y=g|X)}{\mathbb{P}(Y=G|X)} = \beta_{g,0}+\beta_g{}^\top X$$

(The choice of base class G is arbitrary)

They both specify the log-odds as linear models!

Logistic versus LDA

We can write the joint distribution of Y and X as

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

The previous slide shows that $\mathbb{P}(Y|X)$ is the same for both methods:

$$\mathbb{P}(Y = g|X) = \frac{e^{\alpha_{g,0}+\alpha_g^\top X}}{1 + \sum_{k=1}^{G-1} e^{\alpha_{k,0}+\alpha_k^\top X}}$$

- Logistic regression leaves $\mathbb{P}(X)$ arbitrary, and implicitly estimates it with the empirical measure

  (This could be interpreted as a frequentist approach, where we are maximizing the likelihood only and using the improper uniform prior)
- LDA models

$$\mathbb{P}(X, Y = g) = \mathbb{P}(X|Y = g)\mathbb{P}(Y = g) = N(X; \mu_g, \Sigma)\pi_g$$

Logistic versus LDA

Some remarks:

- Logistic regression requires fewer assumptions
- The MLEs under logistic regression will be undefined if the classes are perfectly separable
- If some entries in X are qualitative, then the modeling assumptions behind LDA are suspect
- In practice, the two methods tend to give very similar results