Lecture 4

DJM

16 October 2018

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 $Y \in \mathcal{K} = \{1, \ldots, K\}$

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

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

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

- zeros on the diagonals
- $\ell(k, k')$ on the off diagonal ($k \neq k'$)

How do we measure quality?

Again, we appeal to risk

$$R(g) = \mathbb{E}_{(X,Y)}\ell(Y, g(X))$$

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

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

This can be minimized point wise over X, to produce

$$g_*(X) = \arg\min_{f\in\mathcal{G}} \sum_{y=1}^{K} \ell(y, g(X))\mathbb{P}(Y = y|X)$$

(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

Define (for convenience)

$$\ell_g(Z) = \ell(Y, g(X))$$

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

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

Best classifier

Suppose we encode a two-class response 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

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

(note that $f_* \in [0, 1]$)

Hence, we achieve the same Bayes' rule/limit with squared error classification by discretizing the probability:

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

Classification is easier than regression

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

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

It can be shown that

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

~~Can you show this?~~

Classification is easier than regression

Now

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

Therefore

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

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 $\widehat{g} \in \Gamma$ 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 $\widehat{f}$ and plug it in to the Bayes' rule
- **Density estimation:** Estimate $\widehat{\pi}$ and f_g from $\mathcal{D}$ where $Y = g$ and

Linear classifiers

Linear classifier

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

The boundaries between these regions are known as **decision boundaries**

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

A **linear classifier** is a $\widehat{g}$ that produces linear decision boundaries (don't confuse this with "linear smoothers", regression functions which give $\widehat{Y} = HY$ for some H)

Linear classifier: Example

Suppose $\mathcal{G} = \{0, 1\}$ and we form the GLM logistic regression

The posterior probabilities are

$$\mathbb{P}(Y = 1|X) = \frac{\exp\{\beta_0 + \beta^\top X\}}{1 + \exp\{\beta_0 + \beta^\top X\}}$$
$$\mathbb{P}(Y = 0|X) = \frac{1}{1 + \exp\{\beta_0 + \beta^\top X\}}$$

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

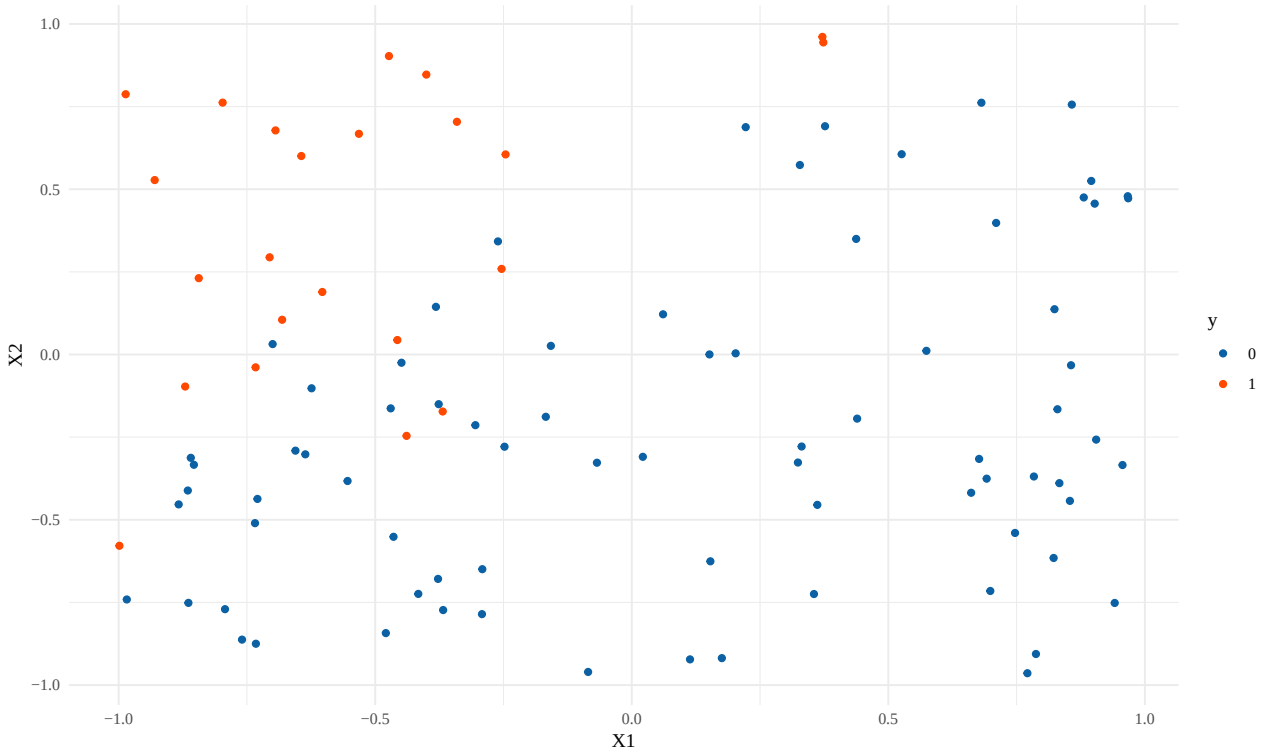
$$\log\left(\frac{\mathbb{P}(Y = 1|X)}{\mathbb{P}(Y = 0|X)}\right) = \beta_0 + \beta^\top X$$

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

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

Logit example

```
g <- ggplot(df, aes(X1,X2,color=y)) + geom_point() +
  scale_color_manual(values=c(blue,red))
g
```




```
log.lm = glm(y~.,data=df, family='binomial')
summary(log.lm)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.63909  -0.24256  -0.03246  -0.00061   2.35104
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.803      1.010  -3.764 0.000167 ***
## X1            -6.593      1.918  -3.438 0.000586 ***
## X2             7.271      1.945   3.739 0.000185 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 107.855  on 99  degrees of freedom
## Residual deviance:  36.785  on 97  degrees of freedom
## AIC: 42.785
##
## Number of Fisher Scoring iterations: 8
```

What is the line?

- Suppose we decide "Predict `1` if `predict(log.lm) > 0.5`".
- This means "For which combinations of `x1` and `x2` is

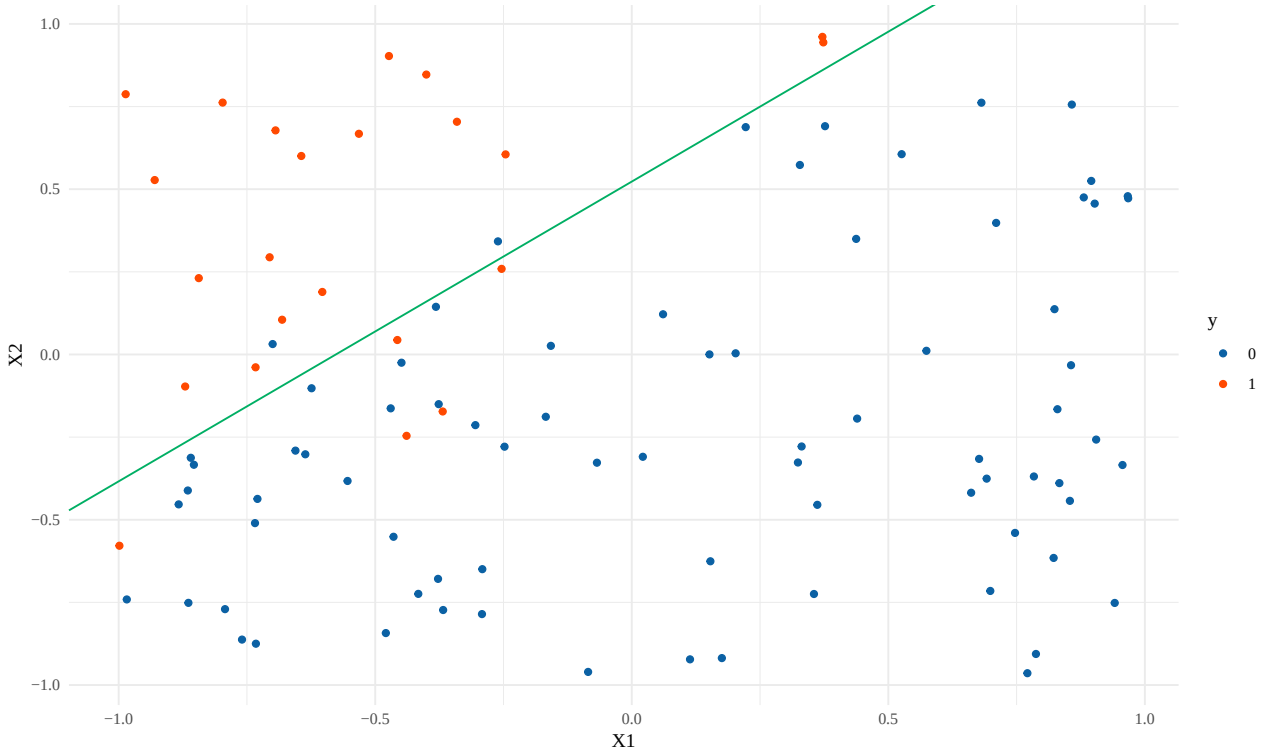
$$\frac{\exp\left(\widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2\right)}{1 + \exp\left(\widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2\right)} > 0.5?$$

- Solving this gives

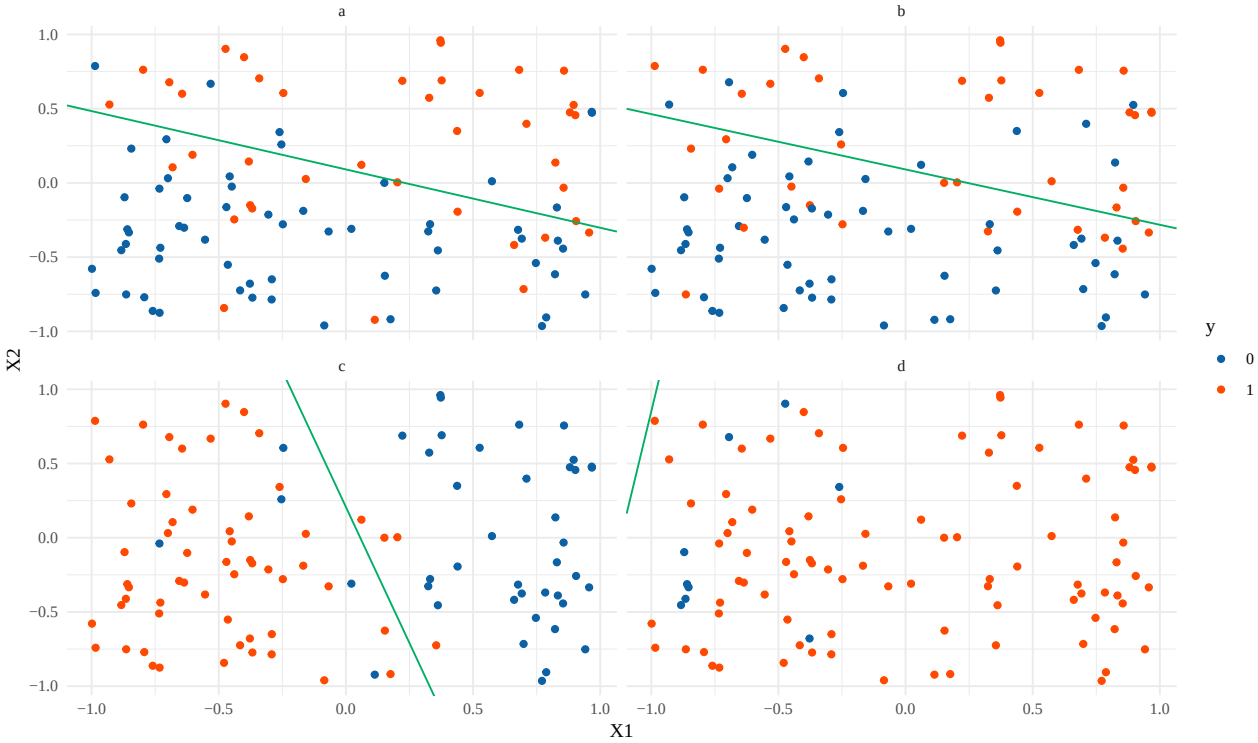
$$\widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 > \log(.5) - \log(1 - .5) \Rightarrow x_2 > -\frac{\widehat{\beta}_0 + \widehat{\beta}_1 x_1}{\widehat{\beta}_2}.$$

- That's just a line. Let's plot it:

```
cc = coefficients(log.lm)
g + geom_abline(intercept = -cc[1]/cc[3], slope = -cc[2]/cc[3], color=green)
```



Lots of different boundaries




```
## # A tibble: 4 x 3
## # Groups:   index [4]
##   index intercept  slope
##   <chr>     <dbl>  <dbl>
## 1 a        0.0907 -0.393
## 2 b        0.0905 -0.373
## 3 c        0.207  -3.67
## 4 d        7.96    7.11
```

Using Bayes Rule

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 covariates 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

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

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 $g(X) = \arg\min_g \delta_g(X)$

(This is just minimum Euclidean distance, weighted by the covariance matrix and prior probabilities)

LDA parameter estimation

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

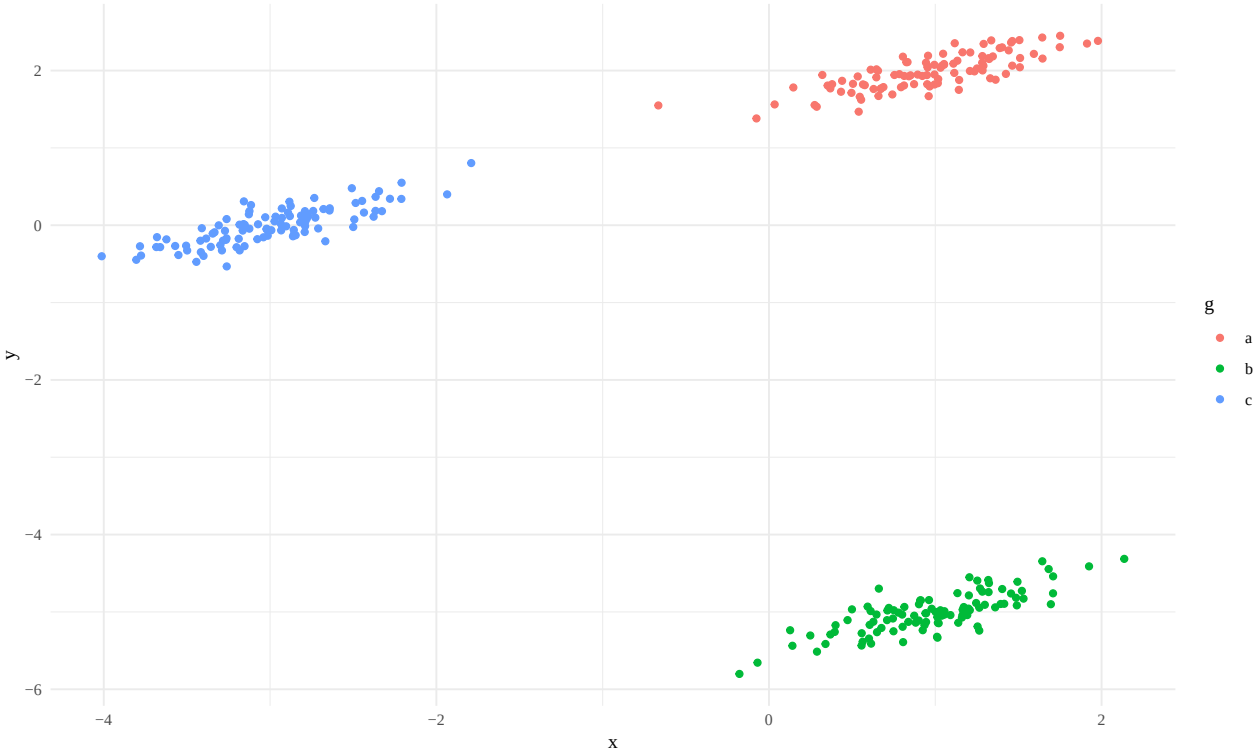
- use the intuitive estimators $\widehat{\mu}_g = \overline{X}_g$ and

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

  then we have produced ~~linear discriminant analysis~~ (LDA)

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

$$\begin{aligned}\tilde{g}(X) &= \arg\min_g \left\|X - \overline{X}_g\right\|_2^2 \\ &= \arg\min_g X^\top X - 2X^\top \overline{X}_g + \overline{X}_g^\top \overline{X}_g \\ &= \arg\min_g -X^\top \overline{X}_g + \frac{1}{2}\overline{X}_g^\top \overline{X}_g\end{aligned}$$

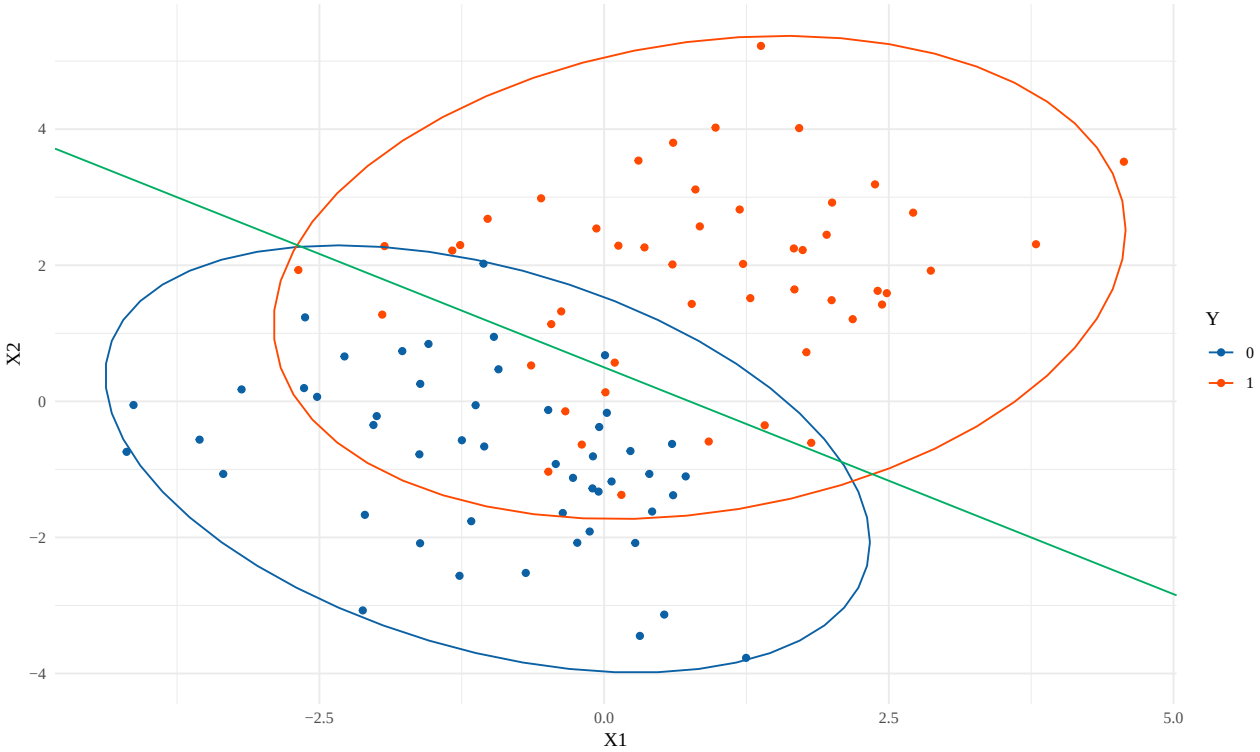
compare this to:

$$\widehat{g} = \arg\min_g \underbrace{X^\top \widehat{\Sigma}^{-1}\overline{X}_g - \frac{1}{2}\overline{X}_g^\top \widehat{\Sigma}^{-1}\overline{X}_g}_{likelihood} + \underbrace{\log(\widehat{\pi}_g)}_{prior}$$

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

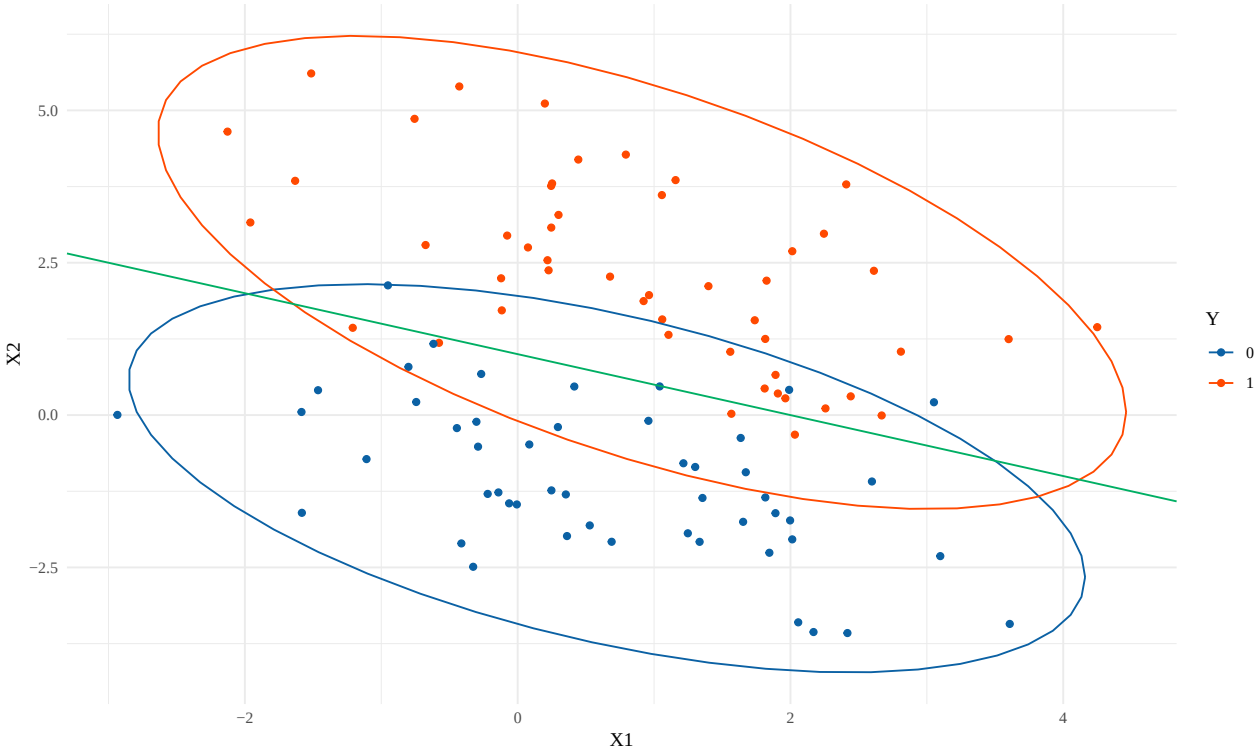
(Note: this generalization of Euclidean distance is called **Mahalanobis** distance)

Intuition



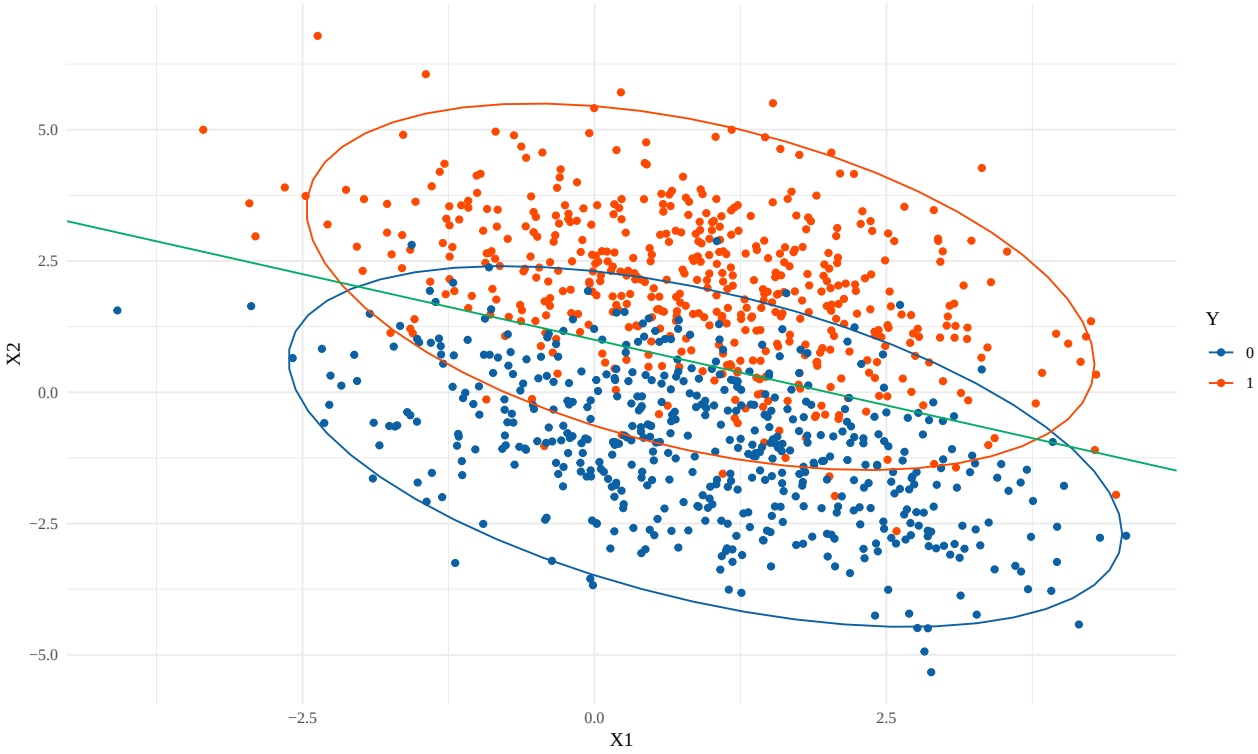
- Note: here there is a single Σ, but I don't know how to plot ellipses in `ggplot`. So these are estimated.

Intuition

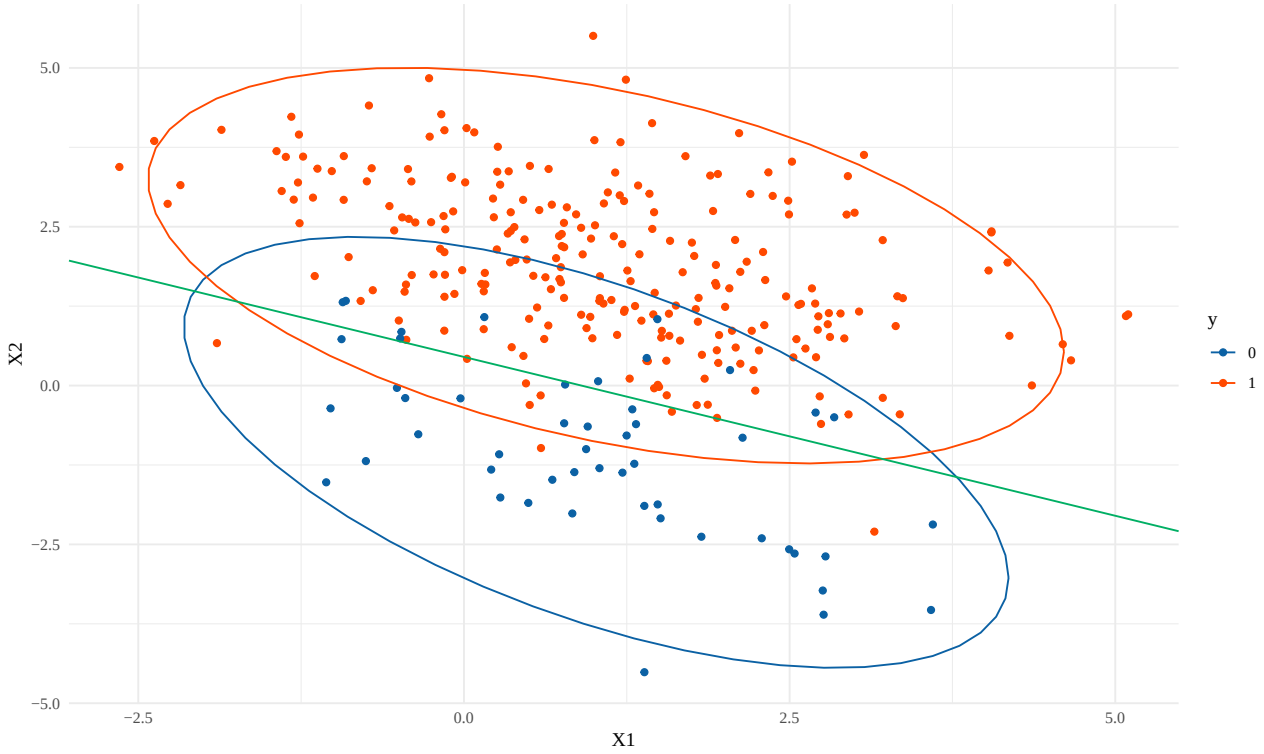


- Different μ_g, Σ.

Same one, but make n big



Same one, but change P(Y=1)



Performance of LDA

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

- The sample size n

  (This determines how accurate the $\widehat{\pi}_g$, $\widehat{\mu}_g$, and $\widehat{\Sigma}$ 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

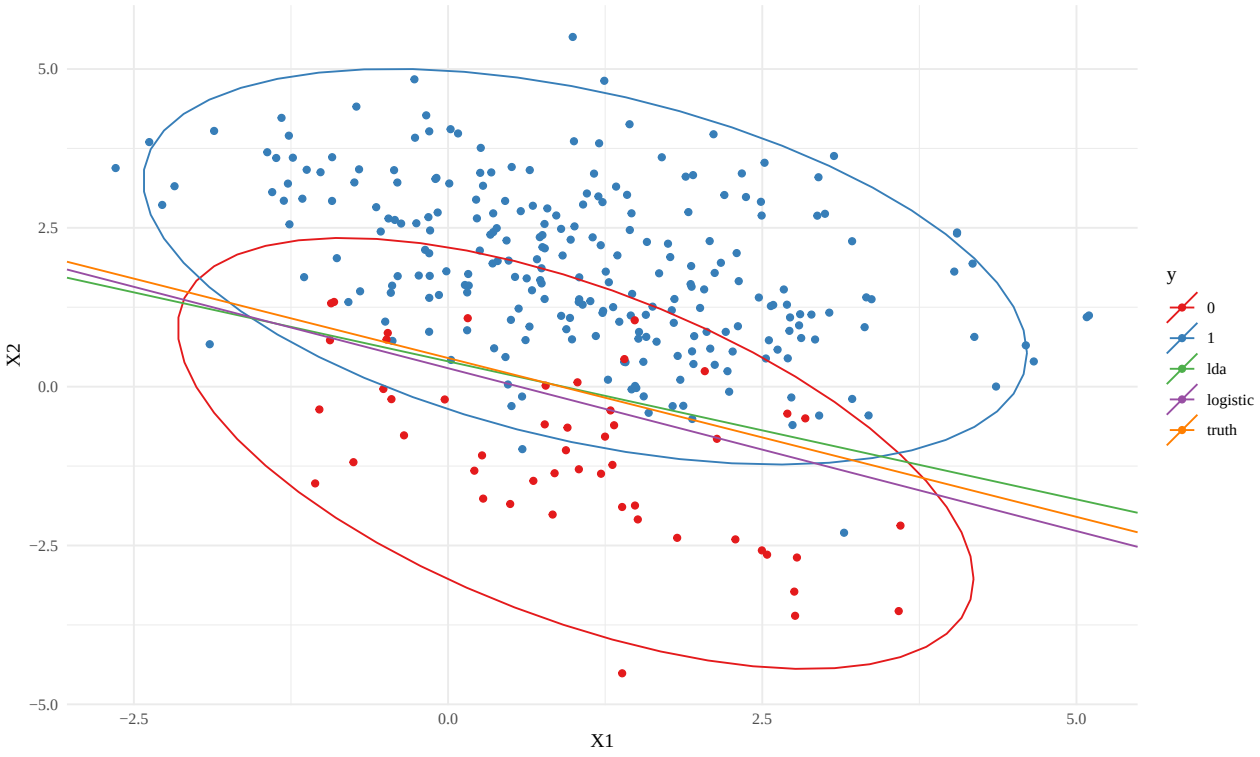
Comparing LDA and Logistic regression

- Both are linear in x:
  - LDA$\longrightarrow \alpha_0 + \alpha_1^\top x$
  - Logit$\longrightarrow \beta_0 + \beta_1^\top x$.
- But the parameters are estimated differently.
- Examine the joint distribution of (X, y):

- LDA: $\prod_i f(x_i, y_i) = \underbrace{\prod_i f(X_i|y_i)}_{\text{Gaussian}} \underbrace{\prod_i f(y_i)}_{\text{Bernoulli}}$
- Logistic: $\prod_i f(x_i, y_i) = \underbrace{\prod_i f(y_i|X_i)}_{\text{Logistic}} \underbrace{\prod_i f(X_i)}_{\text{Ignored}}$

- LDA estimates the joint, but Logistic estimates only the conditional distribution. But this is really all we need.
- So logistic requires fewer assumptions.
- But if the two classes are perfectly separable, logistic crashes (and the MLE is undefined)
- LDA works even if the conditional isn't normal, but works poorly if any X is qualitative

Comparison



LDA family

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)

Regularizing LDA

- Penalize the 'plug-in' estimates (Friedman 1989):
  Let $\lambda \in [0,1]$,
  $$\widehat{\Sigma}_\lambda = \lambda\widehat{\Sigma} + (1-\lambda)\widehat{\sigma}^2 I$$
- Nearest Shrunken Centroids:
  Take $\widehat{\Sigma} = \widehat{\sigma}^2 I$.
- Regularized Optimal Affine Discriminant (ROAD) (Fan, Feng, and Tong 2012):
  Solve
  $$\arg\min_v v'Sv + \lambda \|v\|_1, \qquad \text{s.t. } v'(\widehat{\mu}_1 - \widehat{\mu}_2) = 1.$$
  The intuition is that the discriminant
  $$\widehat{\Sigma}^{-1}(\widehat{\mu}_1 - \widehat{\mu}_2)$$
  now depends only on a few features. This one works really well in high-dimensions.

QDA

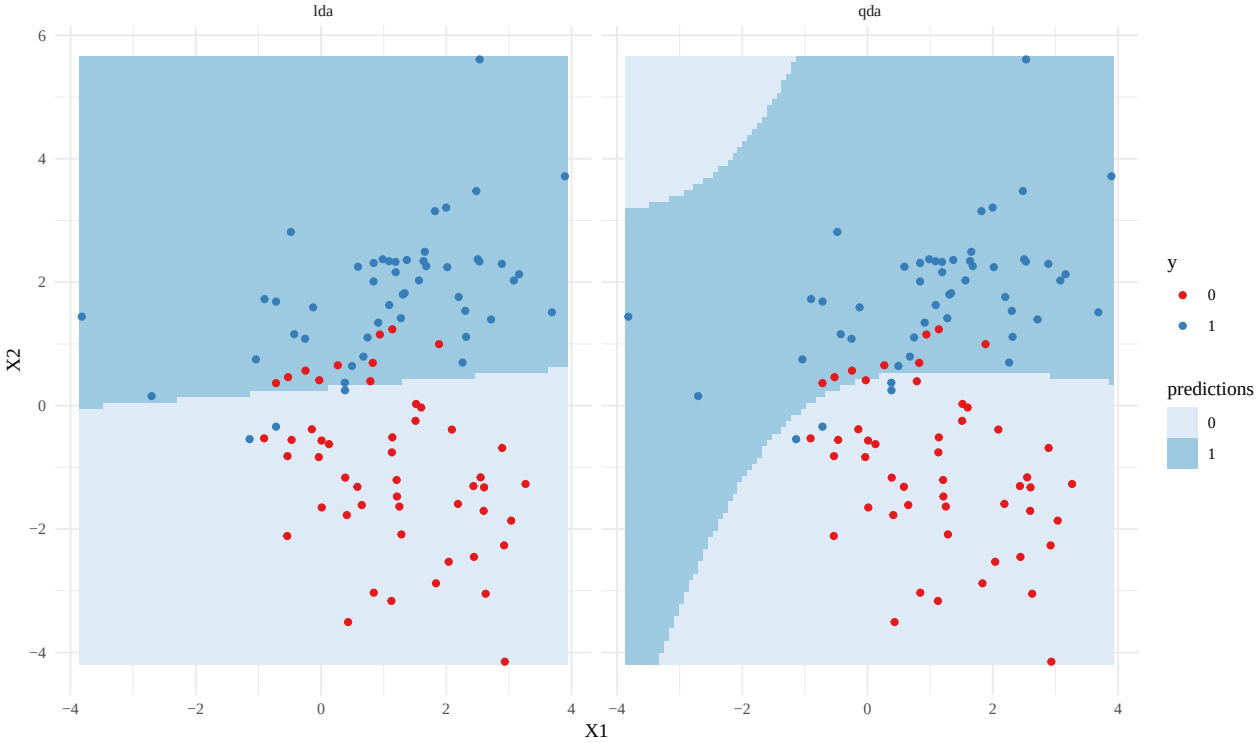
QDA

- Start like LDA, but let $\Sigma_1 \neq \Sigma_0$.
- This gives a "quadratic" decision boundary (it's a curve).
- If we have many columns in X (p)
  - Logistic estimates $p+1$ parameters
  - LDA estimates $2p + p(p+1)/2 + 1$
  - QDA estimates $2p + p(p+1) + 1$
- If $p = 50$,
  - Logistic: 51
  - LDA: 1376 (estimating the discriminant is only 51)
  - QDA: 2651
- QDA doesn't get used much: there are better nonlinear versions with way "fewer" parameters (SVMs)

LDA/QDA in R

```
# Generate data
n1=50; n0=50
Sigma1 = matrix(c(2,.8,.8,1),2)
Sigma0 = matrix(c(1,-.5,-.5,2),2)
X1 = rmvnorm(n1, mu1, Sigma1)
X2 = rmvnorm(n0, mu0, Sigma0)
X = rbind(X1,X2)
y = factor(c(rep(1,n1),rep(0,n0)))
df = data.frame(y,X)
library(MASS)
qda.fit = qda(y~X1+X2, data=df)
lda.fit = lda(y~X1+X2, data=df)
```

LDA/QDA decision boundaries



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 $\widehat{\Sigma}_\lambda = UDU^\top$
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 further 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 $\widehat{\Sigma}$ as the common covariance matrix
3. ~~Sphere:~~ $\tilde{M} = M\widehat{\Sigma}^{-1/2}$
4. ~~Between Covariance:~~ Find covariance matrix for $\tilde{M}$, call it B
5. ~~Spectrum:~~ Compute $B = VSV^\top$

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

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

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** (when $L = 1$)

Logit family

Logistic regression

Logistic regression for two classes simplifies to a likelihood:

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

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

This gets optimized via Newton-Raphson updates and iteratively reweighed 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 minimizing (over β_0, β):

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

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

This is the **logistic elastic net**

Sparse logistic regression: Software

`glmnet` works essentially as before:

```
cv.glmnet(x,y, family='binomial')
```

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

`glmpath` does quadratic approximations to the profiles, while still computing the exact points at which the active set changes

If $G > 2$, use multinomial logit

```
cv.glmnet(x,y, family='multinomial')
```

or

```
library(nnet)
multinom(y~., data=df)
```

Final words: Logistic versus LDA

- Forming **logistic** requires fewer assumptions
- The MLEs under **logistic** 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

Selected references

Fan, Jianqing, Yang Feng, and Xin Tong. 2012. "A Road to Classification in High Dimensional Space." *Journal of the Royal Statistical Society. Series B, Statistical Methodology* 74 (4): 745–71.

Friedman, Jerome H. 1989. "Regularized Discriminant Analysis." *Journal of the American Statistical Association* 84 (405): 165–75.